Journal of Computational and Applied Mathematics 473 (2026) 116876

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Bi-objective optimization of human-robot collaborative mixed-model multiple assembly lines considering model assignment and energy consumption

Oktay Yilmaz [a], Nezir Aydin [a,b,*], Ibrahim Kucukkoc [c]

[a] Yildiz Technical University, Department of Industrial Engineering, Istanbul, Türkiye
[b] College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
[c] Balikesir University, Department of Industrial Engineering, Balikesir, Türkiye

ARTICLE INFO

Keywords:
Multiple assembly lines
Model-line assignment
NSGA-II
Mixed-model
Energy consumption
Collaborative robots

ABSTRACT

A critical yet often overlooked challenge in mixed-model and multi-line production environments is the model-line assignment problem–deciding which product models should be allocated to which assembly lines. This decision has a profound effect on overall production efficiency, as it directly influences subsequent balancing and scheduling decisions. The integration of collaborative robots (cobots) into these environments further complicates this task. Despite its significance, the joint consideration of model-line assignment and robotic line balancing has received limited attention in the literature. This study addresses this gap by formulating the robotic mixed-model multiple assembly line balancing problem with simultaneous model-line assignment (MLA-RMMALB) and proposing a multi-objective mixed-integer programming model. The model aims to minimize total production costs and $PM_{2.5}$ emissions resulting from cobots' energy consumption. To handle the complexity of the problem, a Non-dominated Sorting Genetic Algorithm II (NSGA-II) is developed as a solution approach. The model's effectiveness is demonstrated through a numerical example involving 21 tasks and benchmark problems from the literature. Solutions obtained under integrated model-line assignment are compared with random assignment scenarios, revealing significant performance gains in both objectives. NSGA-II proves capable of delivering optimal or near-optimal solutions efficiently for small- and medium-sized instances, and high-quality results for larger problems. This study contributes to literature by addressing critical challenges in multi-line mixed-model production by jointly considering model-line assignment, cobot heterogeneity, and the parallel operation of cobots and human workers. It proposes NSGA-II as an effective solution method for this complex problem. Practically, the study provides a decision-support tool for manufacturers aiming to optimize both cost-efficiency and environmental performance in robotic assembly systems. The findings are especially relevant for industries adopting cobots in high-variety production environments where these factors must be simultaneously managed.

* Corresponding author at: College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
E-mail addresses: oktay.yilmaz@std.yildiz.edu.tr (O. Yilmaz), naydin@hbku.edu.qa (N. Aydin), ikucukkoc@balikesir.edu.tr (I. Kucukkoc).

https://doi.org/10.1016/j.cam.2025.116876
Received 18 January 2025; Received in revised form 16 June 2025;
Available online 21 June 2025

1. Introduction

The optimization of flow-oriented mass production activities has been a critical research area for centuries. Assembly lines, which consist of a series of workstations connected by conveyor belts, have played a key role in achieving efficient production [1,2]. These lines are widely used in various industries, manufacturing products ranging from small consumer goods to large-scale items such as automobiles and aircraft [3]. The challenge of optimally assigning tasks to workstations while considering precedence constraints is commonly referred to as the Assembly Line Balancing Problem (ALBP).

The mathematical formulation of ALBP was first introduced by Salveson [4], focusing on the allocation of tasks to workstations [5]. Since then, numerous studies have expanded upon this foundation, incorporating exact algorithms, heuristics, and metaheuristic approaches [6,7]. These studies address various aspects of assembly line balancing, classifying problems based on industrial objectives, product variety, line configuration, and task durations [8,9].

With advancements in manufacturing technologies, assembly lines have increasingly incorporated multiple assembly line systems, allowing different product models to be produced across independent lines. The transition from single to multiple assembly lines has been influenced by the growing emphasis on flexibility and efficiency in production. While multiple assembly line systems play an important role in modern manufacturing, research on their balancing and model assignment has not been as widely explored. Many existing studies focus on predefined model assignments for each line, with fewer addressing dynamic model-line assignments and mixed-model production.

Although there is extensive research on assembly line balancing, studies specifically examining multiple assembly line balancing remain relatively scarce. Most existing approaches assume fixed model assignments per line, limiting their applicability to dynamic production environments. Furthermore, with the rise of technological innovations and the integration of robots into assembly lines, the objectives of assembly line balancing problems have diversified. These now include minimizing energy consumption, balancing workloads across workstations, reducing idle time, lowering costs, ensuring ergonomic balance, and maximizing line efficiency. Addressing these aspects requires a broader optimization framework that simultaneously considers model-line assignment and assembly line balancing. A schematic representation of a human-robot collaborative multiple assembly line configuration is given in Fig. 1.

The technological evolution in manufacturing has also led to the integration of robots into production systems. Robots offer benefits such as enhanced precision, improved safety, and increased adaptability in performing various assembly tasks [10]. Among these, collaborative robots (cobots) stand out due to their ability to work alongside human operators. Cobots contribute to reducing task duration uncertainties and enhancing production flexibility, making their integration increasingly attractive for modern assembly lines [11]. However, challenges such as energy consumption and environmental impact, particularly $PM_{2.5}$ emission arises with their use.

Given the increasing adoption of cobots and the growing complexity of multiple assembly line systems, there is a notable gap in the

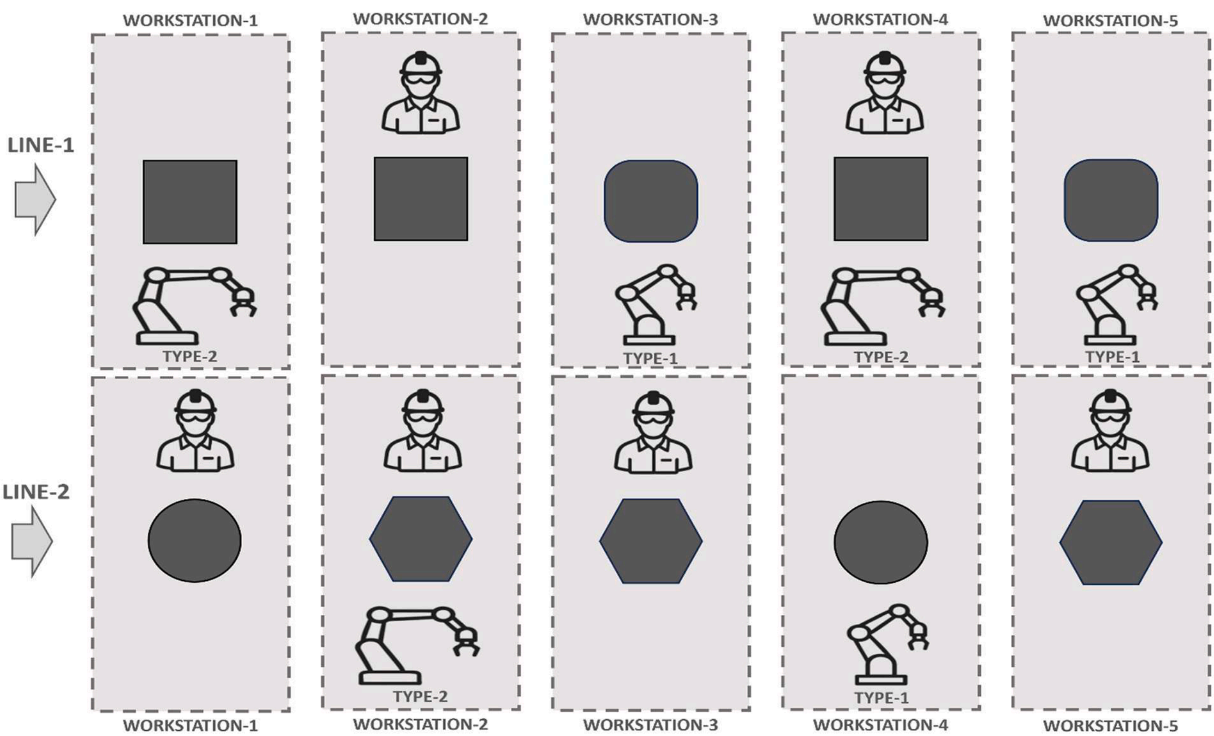

Fig. 1. The schematic representation of a human-robot collaborative multiple assembly line configuration.

literature addressing their combined optimization. Current studies seldom explore the simultaneous integration of model-line assignment, assembly line balancing, and cobot deployment while considering multiple objectives. This research aims to bridge this gap by formulating a comprehensive optimization model that incorporates both the model-line assignment and multiple assembly line balancing problems. The objectives of this study are: (i) to minimize total costs, including workstation and line opening costs, as well as costs associated with cobots and operators, and (ii) to minimize the total $PM_{2.5}$ emission arising from cobot energy consumption.

The contributions of this study can be summarized as follows:

- This study addresses the model-line assignment problem together with the mixed-model multiple assembly line balancing problem.
- It proposes a novel optimization approach for robotic multiple assembly lines, simultaneously minimizing production costs and total $PM_{2.5}$ emission arising from the energy consumption of cobots (multi-objective).
- Non-Dominated Sorting Genetic Algorithm II (NSGA-II) has been effectively developed and introduced as an alternative to exact solution methods, demonstrating its capability in solving complex multi-objective optimization problems.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature, emphasizing studies pertinent to the current research. Section 3 introduces the mathematical model, detailing the objectives and constraints. Section 4 describes the experimental tests, NSGA-II methodology, and results. Finally, Section 5 concludes the study, summarizing key findings and implications.

2. Literature review

Assembly lines are widely utilized as flow-oriented production systems to manufacture high-quality, low-cost, standardized homogeneous products [12]. To determine the research focus, the "Future Directions" sections of survey papers were reviewed to identify gaps in the literature [6,8,13–19]. Aguilar et al. [6] emphasize several research gaps in their "Future Directions" section, including further studies on assigning models to production lines, particularly in mixed models and addressing heterogeneous operators and resource constraints such as robots and equipment. They also advocate for increased research on energy efficiency and semi-automatic workstations that facilitate collaboration between cobots and human operators.

Building on these gaps, this study investigates the model-line assignment problem and the mixed-model multiple robotic assembly line balancing problem, where multiple model types are assigned across multiple assembly lines. No prior research has simultaneously addressed model-line assignment, mixed-model production, and cobot integration in this context. The utilization of cobots is essential due to their ability to enhance flexibility and ergonomic conditions. However, their inclusion complicates task assignment, scheduling, and resource allocation [14,15,20–23].

Examining cobot-related studies reveals different optimization approaches. Levitin et al. [24] and Gao et al. [25] minimized cycle time in single-model robotic assembly lines with heterogeneous and homogeneous robots, respectively. Weckenborg and Spengler [26] focused on cost minimization in production systems using cobots, aligning with this study's objective of minimizing total costs, including line and station opening costs, as well as cobot and operator costs. Weckenborg et al. [27] further investigated single-model straight assembly lines with a limited number of cobots, emphasizing the complexity introduced by cobot-operator collaboration. Sikora and Weckenborg [28] extended this by incorporating the Benders' decomposition algorithm for performance comparison. Cil et al. [29] addressed cycle time minimization but restricted cobot-operator interaction. Boschetti et al. [30] optimized task assignments based on resource capabilities, allowing for task-dependent processing times.

Regarding cobot energy consumption, several studies propose optimization models. Ab Rashid et al. [31] minimized cobot energy use in a single-model straight assembly line, assuming standby energy consumption at 10 % of working-phase energy. Similarly, Belkharroubi and Yahyaoui [32] examined mixed-model robotic assembly lines with heterogeneous robots, while Chitsrisakda et al. [33] explored optimization strategies for robotic mixed-model assembly line balancing problems. Chutima and Khotsaenlee [34] extended this to a mixed-model parallel U-type assembly line, integrating multiple operator types and optimizing five objectives, including cobot energy consumption. Their model is utilized in this study to minimize $PM_{2.5}$ emission arising from cobot energy consumption. Soysal Kurt et al. [10] and Huang et al. [35] also addressed cobot energy use in multi-objective optimization frameworks, with Huang et al. [35] integrating ergonomic risk and cycle time considerations.

The probability of obtaining the optimal solution in assembly line balancing problems is generally expressed as $I!/2^P$, where I represents the number of tasks and P denotes the number of constraints. Consequently, assembly line balancing problems are categorized as NP-HARD problems [36,37]. To address such problems, various heuristic and metaheuristic approaches have been employed [38]. Among the heuristic methods are the ranked positional weight technique (RPWT) [39], LBHA [40], COMSOAL [41], and critical path method (CPM) [42]. Prominent metaheuristic algorithms include the genetic algorithm (GA) [43,44], extreme multi-objective genetic algorithm (EMOGA) [45], multi-objective evolutionary algorithm (MOEA) [46], simulated annealing (SA) [47,48], tabu search (TS) [49,50], artificial bee colony [51], ant colony optimization (ACO) [52–54], variable neighborhood search [55], migrating bird optimization (MBO) [56] and particle swarm optimization (PSO) [57,58]. Metaheuristic methods are particularly regarded as effective tools for solving assembly line balancing problems. Among these, the genetic algorithm stands out as one of the most widely utilized approaches in the literature due to its ability to perform controlled random searches to identify optimal solutions [59–69].

The non-dominated sorting genetic algorithm (NSGA-II) is a prevalent method for solving multi-objective optimization problems [70]. Various studies have applied NSGA-II, including Rabbani et al. [71] on U-type robotic assembly line balancing, Li et al. [72] on robotic assembly lines, minimizing cycle time and purchasing costs, and Guo et al. [73] on stochastic assembly and disassembly balancing. Xu et al. [74] applied NSGA-II to a mixed U-type disassembly line balancing problem, while Farsi et al. [75] combined it with MODA to minimize energy consumption, makespan, and tardiness in parallel U-type assembly lines.

Despite extensive research on assembly line balancing, studies on multiple assembly line balancing remain limited. Most assume pre-determined model-line assignments rather than optimizing them. Scholl and Boysen [76] proposed an algorithm assigning models to parallel assembly lines, but each line could only produce one model type. No existing research considers an optimization-based model-line assignment with mixed-model production across multiple lines.

This study contributes to the literature by introducing a model-line assignment framework where mixed models are produced on multiple lines, heterogeneous cobots collaborate with operators at the same workstation, and both total production costs and $PM_{2.5}$ emission from cobot energy consumption are minimized. Additionally, developing NSGA-II as an alternative to exact solution methods enhances its originality and practical relevance. A summary of the literature is presented in Table 1.

Table 1
Summary of the literature.

Author(s) (Year)	Line Layout	Model Variety	Model-Line Assignment	Objective Function	Robotic	Energy Consumption by Robots	Heuristics (*)
Levitin et al. [24]	Straight	Single	—	Cycle time	X	—	GA
Gao et al. [25]	Straight	Single	—	Cycle time	X	—	hGA
Scholl and Boysen [76]	Parallel	Single	X	Station number	—	—	—
Rabbani et al. [71]	U-Type	Mixed	—	Cost, cycle time	X	—	NSGA-II, MOPSO
Rabbani et al. [77]	Parallel	Mixed	—	Station number, workload smoothness	—	—	NSGA-II, MOPSO
Babazadeh et al. [36]	U-Type	Single	—	Station number, cycle time	—	—	MOGA
Babazadeh and Javadian [37]	U-Type	Single	—	Station number, cycle time	—	—	NSGA-II
Weckenborg and Spengler [26]	Straight	Single	—	Cost	X	X	—
Cil et al. [78]	Straight	Mixed	—	Cycle time	X	—	BA, ABC
Weckenborg et al. [27]	Straight	Single	—	Cycle time	X	—	hGA
Boschetti et al. [30]	Straight	Single	—	Makespan	X	—	—
Li et al. [72]	Straight	Single	—	Cycle time, cost	X	—	NSGA-II, IMABC
Ab Rashid et al. [31]	Straight	Single	—	Smoothness index, energy	X	—	—
Belkharroubi and Yahyaoui [32]	Straight	Mixed	—	Energy	X	X	MBCSA
Chutima and Khotsaenlee [34]	Parallel U-Type	Single	—	Efficiency, energy, energy expenditure among workers, tax deduction	X	X	NSTLBO III
Guo et al. [73]	Parallel	Single	—	Cost, workload smoothness	—	—	VNS-NSGA II
Nourmohammadi et al. [22]	Straight	Single	—	Cycle time, resource number	X	—	SA
Keshvarparast et al. [20]	Multi-Straight	Mixed	—	Cycle time	X	—	—
Mao et al. [21]	U-Type	Single	—	Cycle time	X	—	SA, GA
Sikora and Weckenborg [28]	Straight	Single	—	Cycle time	X	—	GA
Xu et al. [74]	U-Type	Mixed	—	Station number, hazard index, disassembled components, workload smoothness	—	—	NSGA-II
Chitsrisakda et al. [33]	Parallel	Single	—	Efficiency, energy, energy expenditure load variance	X	X	NSGA-III
Farsi et al. [75]	Parallel U-Type	Mixed	—	Makespan, energy, tardiness	X	X	MODA, NSGA-II
Huang et al. [35]	Straight	Mixed	—	Cycle time, energy, energy expenditure of workers	X	X	MDABCSB
Soysal-Kurt et al. [10]	Parallel	Mixed	—	Cycle time, energy	X	X	MNSGA-II
Zacharia et al. [23]	Straight	Single	—	Cycle time, smoothness index	X	—	—
This Paper	Multi-Straight	Mixed	X	Cost, energy	X	X	NSGA-II

(*) ABC: Artificial bee colony, BA: Bee algorithm, GA: Genetic algorithm, hGA: Hybrid genetic algorithm, IMABC: Improved multi-objective artificial bee colony, MBCSA: Memory-based cuckoo search algorithm, MDABCSB: Multi-objective discrete artificial bee colony algorithm with specialist bees, MNSGA: Modified non-dominated sorting genetic algorithm, MODA: Multi-objective dragonfly algorithm, MOGA: Multi Objective Genetic Algorithm, MOPSO: Multi-objective particle swarm optimization, NSGA: Non-dominated sorting genetic algorithm, NSTLBO: Non-dominated sorting teaching-learning based optimization, SA: Simulated annealing, VNS: Hybrid variable neighborhood search

3. Methodology

This study presents a novel approach to the simultaneous model-line assignment and robotic mixed-model multiple assembly line balancing (MLA-RMMALB) problem. The objective is to produce mixed models concurrently on multiple lines while minimizing production costs and $PM_{2.5}$ emission.

3.1. Problem statement

This study introduces a simultaneous MLA-RMMALB problem aimed at producing multiple models on several independent assembly lines while minimizing cost and $PM_{2.5}$ emission. Unlike parallel assembly lines, the multiple assembly lines in this study are completely separate, with no shared workstations or connections between them.

The assumptions of this research are defined as follows:

- The solution approach assigns two or more similar models (mixed-model) to two or more assembly lines in accordance with the objective functions.
- All lines are balanced to operate simultaneously. The number of workstations used on each line in the solution of the objective function, which focuses on the cost of the lines, is used as the upper limit in the solution of subsequent objective functions.
- Each model must be produced according to demand within a certain period of time. Model-line assignments must be made to meet these quantities.
- A common precedence relationship diagram is used for models produced on the same line.
- Operators are homogeneous, but cobots are heterogeneous. These resources are assigned to workstations in accordance with the objective function.
- Not all types of resources can perform all tasks; Therefore, there is a capability matrix that defines which resources can execute specific tasks. Additionally, the duration of tasks may vary depending on the resources' capabilities.
- All tasks must be executable by at least one available resource type (operators and/or cobots).
- The processing times of the tasks are deterministic and known for resource types.
- Tasks cannot be assigned to more than one workstation or resource. Therefore, each task can be assigned to only one resource at one workstation.
- At most, one cobot and one operator can be assigned to a workstation. Therefore, operators and cobots can work together simultaneously.
- While calculating total cost, line opening cost, workstation opening cost, cobot cost, and operator cost are considered. Since cobots are heterogeneous, the unit cost of each cobot type varies.

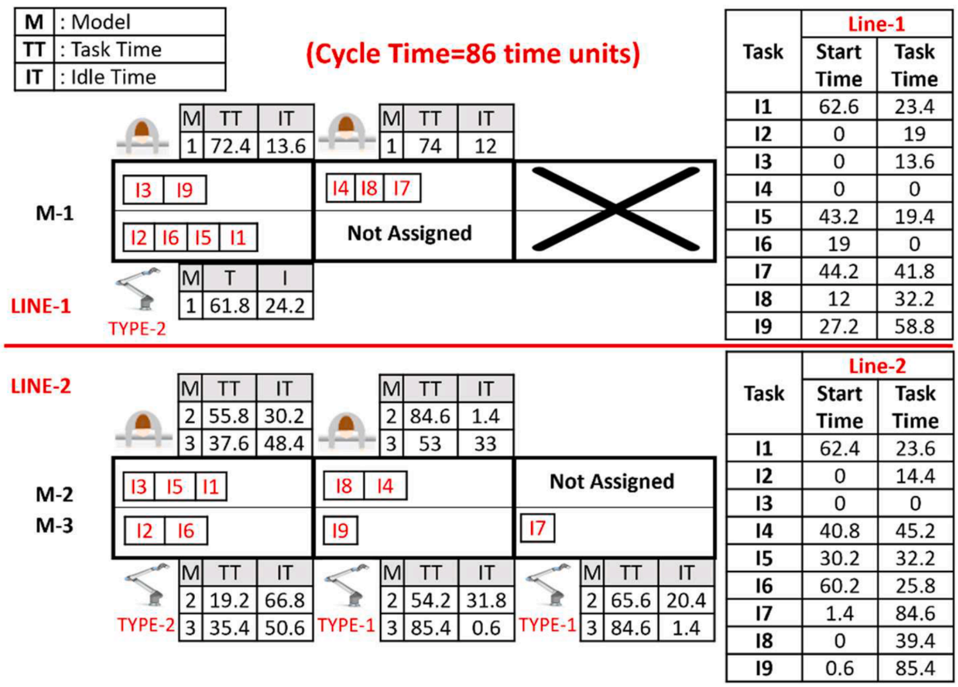

Fig. 2. An illustration of a feasible solution.

3.2. Mathematical model

In the presented problem, a mixed-integer linear programming (MILP) model is developed with two primary objective functions: minimizing line costs and reducing the $PM_{2.5}$ emission arising from the energy consumption of cobots. The mathematical model is structured to address these two objectives separately. In MILP-1, the focus is solely on minimizing the cost of the assembly lines. In MILP-2, the primary concern is reducing the $PM_{2.5}$ emission arising from the energy consumption of cobots. The parameters, decision variables, and constraints associated with MILP-1 are introduced at this stage. It is important to note that this paper builds upon the work of Yilmaz et al. [79], extending it by incorporating NSGA-II, comparing various model-line assignment strategies, and conducting a more comprehensive benchmark problem analysis.

Before addressing the multi-objective function, MILP-1 and MILP-2 are initially solved independently. Subsequently, the minimum and maximum values obtained from these solutions are normalized. The normalization process utilizes the formula $(obj - obj_{min}) / (obj_{max} - obj_{min})$ for both objective functions. In the multi-objective function, the importance coefficients of these two objectives are considered equal [80,72]. Fig. 2 illustrates a feasible solution to the proposed mathematical model.

As seen in the figure, there are two independent assembly lines. M-1 is assigned on the first line, while M-2 and M-3 are assigned on the second line, with a cycle time of 86 time units. The first line has two workstations, whereas the second has three. In some workstations, an operator or a cobot is present, while in others, both an operator and a cobot work together. Looking at the second workstation of the second line, tasks 8 and 4 are performed by the operator, while Cobot Type-1 performs task 9. The start time of task 4 assigned to the operator is 40.8, with a task time of 45.2 time units. The task time is the maximum task time defined for that resource among the models assigned to the line. For this operator, the task time for M-2 is 84.6, and the waiting time is 1.6 time units.

3.2.1. MILP-1

In MILP-1, only production line costs are considered. The cost data for workstations, cobots, and operators are sourced from Weckenborg and Spengler [26], who developed a mathematical model to minimize these costs based on cycle time. In their study, operators and cobots are homogeneous and can work in parallel at the same station. In contrast, this paper additionally accounts for reducing the $PM_{2.5}$ emission arising from the energy consumption of cobots and introduces a heterogeneous cobot structure, where different cobot types have varying capabilities and task processing times.

Weckenborg and Spengler [26] generated their cost data under specific assumptions, including a five-year machine lifespan, 230 annual working days, and an eight-hour daily shift. Based on these parameters, the amortization period is calculated as $5 \times 230 \times 8 \times 60 = 552{,}000$ min, with all costs expressed per minute. The total cost of a cobot (including necessary equipment) amounts to 45, 000/552,000 = 0.08 euros per minute, whereas operator (labor) costs are estimated at 35/60 = 0.6 euros per minute, assuming an hourly wage of 35 euros. This analysis reveals that operator costs are approximately 7.5 times higher than cobot costs, prompting the mathematical model to prioritize cobot utilization unless constrained by capability limitations or resource shortages.

Since operators and cobots can perform different tasks simultaneously at the same workstation, ensuring compliance with predecessor-successor relationships poses a challenge. To address this, the model incorporates a decision variable for task start times, guaranteeing that if tasks are assigned to the same workstation but different resources, any predecessor task is completed before its successor begins.

Furthermore, given the presence of multiple cobot types and the complexity of the multi-assembly line problem, alternative cost data for cobots, workstation setups, and assembly line openings are derived to align with Weckenborg and Spengler [26]. Additionally, the number of workstations used in MILP-1 serves as the upper limit for MILP-2 and the multi-objective optimization. The mathematical model for MILP-1 is presented below [79].

3.2.1.1. Sets and indices.

I: Set of tasks $(i, j \in I)$

W: Set of workstations $(w \in W)$

M: Set of models $(m \in M)$

L: Set of assembly lines $(l \in L)$

H: Set of operators $(h \in H)$

C: Set of cobots $(c \in C)$

R: Set of resources $(R = H \cup C)$ and $(h, c,\ r \in R)$

3.2.1.2. Parameters.

cp_{ri}: Capability of resource r to perform task i, which equals 1 if resource r can perform task i; 0, otherwise

$tfrs_{irm}$: Required time for resource r to execute task i for model m

D_m: Demand for model m $(m \in M)$

P_{ij}: Precedence relationship between tasks i and j, where $(i, j) \in P$ if task i is an immediate predecessor of task j

nor_r: Quantity for each type of resource r

cor_r: Cost per unit time of each type of resource r

cos: Cost per unit time of opened workstations

col: Cost per unit time of opened lines

CT: Cycle time

PP: Planning period

3.2.1.3. Decision variables.

x_{iwlr}: 1, if task i is assigned to resource r in workstation w on line l; 0, otherwise

y_{ml}: 1, if model m is assigned to line l; 0, otherwise

z_{wl}: 1, if workstation w on line l is opened; 0, otherwise

ln_l: 1, line l is opened; 0, otherwise

ra_{wlr}: 1, if resource type r is assigned to workstation w on line l; 0, otherwise

ssa_{ijwl}: 1, if task i is before the task j in workstation w on line l; 0, otherwise

st_{il}: Start time of task i on line l

3.2.1.5. Objective function. The first objective function given in Eq. (1) comprises three distinct components (i) the total operational cost of open (utilized) assembly lines, (ii) the total cost of open workstations across all lines, and (iii) the cumulative cost of all deployed resources.

$$obj_1 = \sum_{l\in L}(ln_l \times col \times CT) + \sum_{w\in W}\sum_{l\in L}(z_{wl} \times \cos \times CT) + \sum_{w\in W}\sum_{l\in L}\sum_{r\in R}(ra_{wlr} \times cor_r \times CT) \tag{1}$$

3.2.1.6. Constraints. Constraint (2) ensures that one of each task is assigned to one of the open (utilized) assembly lines.

$$\sum_{w\in W}\sum_{r\in R} x_{iwlr} = ln_l, \ \forall\, i \in I;\ l \in L \tag{2}$$

Constraint (3) prevents tasks from being assigned to resources that are incapable of performing them.

$$x_{iwlr} \le cp_{ri}, \ \forall\, i \in I;\ w \in W;\ l \in L;\ r \in R \tag{3}$$

Constraint (4) ensures that each model is exactly assigned to an assembly line.

$$\sum_{l\in L} y_{ml} \ = 1, \ \forall\, m \in M \tag{4}$$

Constraints (5) and (6) ensure that a line with no model assignment is not opened, whereas a line with at least one model is assigned is opened.

$$ln_l \le \sum_{m\in M} y_{ml}, \ \forall\, l \in L \tag{5}$$

$$y_{ml} \leq ln_l, \ \forall \ m \in M; \ l \in L \tag{6}$$

Constraint (7) prevents opening workstations to which no resources have been assigned.

$$z_{wl} \leq \sum_{r \in R} ra_{wlr}, \ \forall \ w \in W; \ l \in L \tag{7}$$

Constraints (8) and (9) ensure that at most one operator and one cobot are assigned to each workstation on the lines.

$$\sum_{h \in H} ra_{wlh} \leq z_{wl}, \ \forall \ w \in W; \ l \in L \tag{8}$$

$$\sum_{c \in C} ra_{wlc} \leq z_{wl}, \ \forall \ w \in W; \ l \in L \tag{9}$$

Constraint (10) ensures that no resources are present at the workstation unless they are assigned a task suitable for their capabilities.

$$ra_{wlr} \leq \sum_{i \in I} x_{iwlr}, \ \forall \ w \in W; \ l \in L; \ r \in R \tag{10}$$

Constraint (11) ensures that a task is not assigned to a workstation without at least one suitable resource.

$$\sum_{i \in I} x_{iwlr} \leq ra_{wlr} \times |I|, \ \forall \ w \in W; \ l \in L; \ r \in R \tag{11}$$

Constraint (12) ensures that the number of resources of each type used on all lines does not exceed the total number of resources available.

$$\sum_{w \in W} \sum_{l \in L} ra_{wlr} \leq nor_r, \ \forall \ r \in R \tag{12}$$

Constraint (13) ensures that all predecessors of a task must be assigned earlier than that task.

$$\sum_{w \in W} \sum_{r \in R} \left(w \times x_{iwlr} - w \times x_{jwlr} \right) \leq 0, \ \forall \ i,j \in P_{ij}; \ l \in L \tag{13}$$

Constraints (14) and (15) ensure that lines and workstations are opened, respectively.

$$z_{w+1l} \leq z_{wl}, \ \forall \ w \in W; \ l \in L \tag{14}$$

$$ln_{l+1} \leq ln_l, \ \forall \ l \in L \tag{15}$$

Constraints (16) and (17) ensure that the sum of the task start time and its execution time does not exceed the cycle time. They also prevent assigning start times to tasks on lines that are not opened.

$$st_{il} + \sum_{w \in W} \sum_{r \in R} (x_{iwlr} \times tfrs_{irm}) \leq CT \times \ (2 - y_{ml}), \ \forall \ i \in I; \ l \in L; \ m \in M \tag{16}$$

$$st_{il} \leq ln_l \times CT, \ \forall \ i \in I; \ l \in L \tag{17}$$

For tasks with a precedence relationship between them, constraint (18) ensures that the sum of the start time and execution time of the preceding task is less than or equal to the start time of the next task.

$$st_{il} + \sum_{r \in R} (x_{iwlr} \times tfrs_{irm}) \leq st_{jl} + CT \times (1 - y_{ml}) + CT \times \left(1 - ssa_{ijwl}\right), \ \forall \ i,j \in I; \ w \in W; \ l \in L; \ m \in M \tag{18}$$

Constraint (19) ensures that when tasks with a precedence relationship (P_{ij}) in between are assigned to the same workstation, a priority-subsequence relationship (ssa_{ijwl}) exists between them, regardless of the resource to which they are assigned.

$$1 - \left(1 - \sum_{r \in R} x_{iwlr}\right) - \left(1 - \sum_{r \in R} x_{jwlr}\right) \leq ssa_{ijwl}, \ \forall \ i,j \in P_{ij}; w \in W; l \in L \tag{19}$$

Constraint (20) guarantees that a priority-subsequence relationship (ssa_{ijwl}) exists between all tasks assigned to the same resource at the same workstation.

$$1 - (1 - x_{iwlr}) - \left(1 - x_{jwlr}\right) \leq ssa_{ijwl} + ssa_{jiwl}, \ \forall \ i,j \in I, \ i \neq j; w \in W; l \in L; r \in R \tag{20}$$

Constraint (21) ensures that no priority-subsequence relationship (ssa_{ijwl}) exists between tasks that are not assigned to the same workstation.

$$ssa_{ijwl} \le \left(\sum_{r \in R} x_{iwlr} + \sum_{r \in R} x_{jwlr} \right) \Big/ 2, \ \forall \ i,j \in I; w \in W; l \in L \tag{21}$$

Constraint (22) ensures that the sum of the demands of the models assigned to each assembly line is produced in the planning period.

$$\sum_{l \in L} y_{ml} \ \times D_m \le PP \Big/ CT, \ \forall \ l \in L \tag{22}$$

Constraints (23) - (29) identify the type of decision variables.

$$x_{iwlr} \in \{0,1\}, \ \forall \ i \in I, \ w \in W, \ l \in L, \ r \in R \tag{23}$$

$$y_{ml} \in \{0,1\}, \ \forall \ m \in M, l \in L \tag{24}$$

$$z_{wl} \in \{0,1\}, \ \forall \ w \in W, \ l \in L \tag{25}$$

$$ln_l \in \{0,1\}, \ \forall \ l \in L \tag{26}$$

$$ra_{wlr} \in \{0,1\}, \ \forall \ w \in W, l \in L, r \in R \tag{27}$$

$$ssa_{ijwl} \in \{0,1\}, \ \forall \ i,j \in I, w \in W, l \in L \tag{28}$$

$$st_{il} \ge 0, \ \forall \ i \in I, l \in L \tag{29}$$

3.2.2. MILP-2

In MILP-2, the objective function focuses solely on minimizing the $PM_{2.5}$ emission resulting from cobot energy consumption at workstations. The $PM_{2.5}$ emission calculation accounting for both working and standby phases, follows the approach of Chutima and Khotsaenlee [34]. Their study examined parallel U-shaped assembly lines using a multi-objective framework that included $PM_{2.5}$ emissions from cobot energy use. While their model assumed heterogeneous operators and homogeneous cobots, with each workstation assigned either an operator or a cobot, this study extends their work in several key aspects: (1) incorporating production costs for multiple straight assembly lines, (2) introducing heterogeneous cobots with varying capabilities and task processing times, and (3) allowing parallel operation of an operator and cobot at the same workstation. Consistent with their approach, we maintain the assumption that cobot standby energy consumption equals 10 % of working-phase consumption.

Given the presence of multiple cobot types in this study, energy consumption during task execution is determined based on the specific cobot type. It is assumed that cobots that complete a task in a shorter time will consume more energy per unit of time. The parameters, decision variables, and constraints that need to be incorporated into MILP-1 are outlined below.

3.2.2.1. Additional parameters.

OPC_c: Energy consumption for each type of cobot c per unit time during the operation

SPC_c: Energy consumption for each type of cobot c per unit time during standby

$$SPC_c = 0,1 \times OPC_c$$

PMR: Average $PM_{2.5}$ emission coefficient of the energy consumption (g/kWh)

s: A sufficiently small positive number

3.2.2.2. Additional decision variables.

TTM_{lwrm}: Total task time for model m of tasks assigned to line l workstation w resource type r

TIM_{lwrm}: Total idle time for model m of tasks assigned to line l workstation w resource type r

3.2.2.3. Objective function. The second objective function, defined in Eq. (30), minimizes the $PM_{2.5}$ emissions generated by the total energy consumption of various cobot types assigned to workstations. The first component accounts for emissions produced during active task execution, while the second component represents emissions occurring during idle periods.

$$obj_2 = \sum_{l\in L}\sum_{w\in W}\sum_{c\in C}\sum_{m\in M}(TTM_{lwcm} \times OPC_c \times PMR) + \sum_{l\in L}\sum_{w\in W}\sum_{c\in C}\sum_{m\in M}(TIM_{lwcm} \times SPC_c \times PMR) \tag{30}$$

3.2.2.4. Additional constraints. Constraints (31) and (32) determine the total task and idle times of each resource assigned to the lines, for each model.

$$\sum_{i\in I}(x_{iwlr} \times tfrs_{irm}) + \sum_{i\in I}(x_{iwlr} \times tfrs_{irm}) \times (y_{ml} - 1) \le TTM_{lwrm},\ \forall\, l \in L;\ w \in W; r \in R; m \in M \tag{31}$$

$$CT - \sum_{i\in I}(x_{iwlr} \times tfrs_{irm}) + CT \times (y_{ml} - 1) + CT \times (ra_{wlr} - 1) \le TIM_{lwrm},\ \forall\, l \in L;\ w \in W; r \in R; m \in M \tag{32}$$

Constraint (33) ensures that the sum of task and idle times is less than or equal to the cycle time.

$$TTM_{lwrm} + TIM_{l\ wrm} \le CT \times ra_{wlr},\ \forall\, l \in L;\ w \in W; r \in R; m \in M \tag{33}$$

Constraint (34) ensures that no task or idle time is calculated for a model that is not assigned to the line.

$$TTM_{lwrm} + TIM_{lwrm} \le CT \times y_{ml},\ \forall\, l \in L;\ w \in W; r \in R; m \in M \tag{34}$$

Constraint (35) ensures that the total time of the tasks assigned to each resource is greater than zero.

$$\sum_{m\in M} TIM_{lwrm} + s + s \times\ (ln_l - 1) \le \sum_{m\in M} y_{ml} \times CT,\ \forall\, l \in L;\ w \in W; r \in R \tag{35}$$

Constraints (36) and (37) identify the type of decision variables.

$$TTM_{lwrm} \ge 0,\ \forall\, l \in L,\ w \in W, r \in R,\ m \in M \tag{36}$$

$$TIM_{lwrm} \ge 0,\ \forall\, l \in L,\ w \in W, r \in R,\ m \in M \tag{37}$$

4. Experimental tests

Experimental tests of the proposed mathematical model are presented in the following sections. The first part discusses data acquisition and results of two scenarios, each solved using different model-line assignment strategies, providing a detailed explanation of the mathematical model.

To evaluate the model's performance, widely used benchmark problems from the literature are employed. The second part presents the results obtained using the exact solution method applied to these benchmark problems, while the third part compares the results obtained using NSGA-II, an alternative metaheuristic approach to the exact solution method.

4.1. The effect of using the proposed model assignment approach

To assess the impact on the objective functions, models are initially assigned to the lines randomly (in cases where model-line assignment and assembly line balancing are not considered together) in the first scenario (scenario-1). In the second scenario (scenario-2), the model assignments to lines are determined by the solver. A numerical example consisting of 21 tasks is used to demonstrate these effects. For both scenarios, the cycle time is fixed at 200 time units. The configuration includes two assembly lines, each with five feasible workstations. The number of product models is three, and each type of resource is available in a quantity of ten. The problem is first solved separately for each objective function and then for the multi-objective function. One goal of comparing the two scenarios is to observe how the number of assembly lines, workstations, and resources change with this approach. Another objective is to evaluate the sensitivity of the mathematical model by examining the influence of modifying the objective function on the

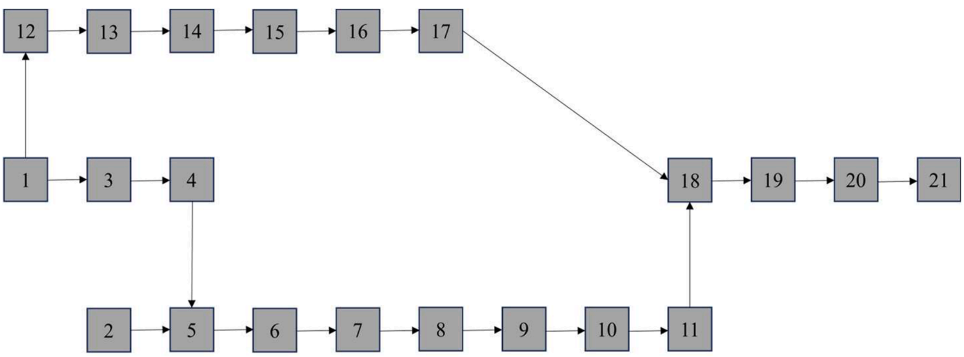

Fig. 3. Common precedence relationship diagram.

prioritized resource type and task assignment strategy.

A sample consisting of three models and 21 tasks, as outlined by Barathwaj et al. [81], is used to create the numerical example. As specified in the sample, the task times for the three models are initially generated for the Operator. Since this study involves two types of cobots, the task times for the cobot types are derived from the times initially generated for the Operator. Information regarding these tasks is provided in Appendix A.

Due to the heterogeneity of the resources, not all resources can perform every task. Therefore, a capability matrix has been developed. Additionally, a common precedence relationship diagram is used to produce the three models. The capability matrix is shown in Appendix B, and the common precedence relationship diagram is presented in Fig. 3.

The data used for calculating the cost (obj_1) and $PM_{2.5}$ emission arising from the total energy consumed by the cobots (obj_2) are presented in Table 2.

The proposed mathematical model coded in Python was solved using Gurobi 11.0.3 on an Intel(R) Core(TM) i9-13900HX CPU running at 2.2 GHz with 64 GB RAM on a Windows 11, 64-bit operating system. The optimal solutions used for normalizing the multi-objective function, obtained by solving MILP-1 and MILP-2 separately, are presented in Table 3. The optimal multi-objective solutions after normalization are displayed in Table 4. Furthermore, the optimal line-balancing solutions for the scenario-1 and scenario-2 are shown in Fig. 4 and Fig. 5.

The results of the scenario-1 and scenario-2, which are created to test the simultaneous MLA-RMMALB problems, are presented above. When the results regarding the numerical example are examined:

- In both scenarios, the mathematical model prioritizes cobots with lower costs when solving MILP-1 (which minimizes the total cost).
- In both scenarios, when solving MILP-2 (which minimizes the total $PM_{2.5}$ emission arising from the energy consumed by cobots), the model assigns tasks to operators as much as possible. The model utilizes all available stations to assign operators the maximum number of tasks.
- Compared to the scenario-1, the scenario-2 achieves better results in both the line costs (obj_1) and total $PM_{2.5}$ emission arising from the consumption by cobots (obj_2). This suggests that the mathematical model effectively identifies the most suitable model-line assignment combinations for its objective functions.
- The solution of obj_{Multi} in the scenario-2, compared to the scenario-1, one less station is opened, and one fewer Type 2 cobot is used. As a result, model-line assignment by the solution approach provides approximately an 8.5 % improvement in cost and a 4 % improvement in $PM_{2.5}$ emission over random assignments.
- Since the solution approach carries out the model-line assignment, the solution time for the scenario-2 is longer than that for the scenario-1.

4.2. Benchmark problems

In addition to scenario-1 and scenario-2, which are used for a detailed explanation of the mathematical model above, benchmark problems in the literature are also used to test the performance of the models (https://assembly-line-balancing.de/). Bowman (8), Jackson (11), Thomopoulos (19), Roszieg (25), Heskia (28), Lutz (32), Gunther (35), Kilbrid (45), Hahn (53), Warnecke (58), Kim (61), Tonge (70), and Arc (83), whose number of tasks gradually increases, are developed in accordance with the problem to produce three or four models.

The times in these benchmark problems (except for Thomopoulos (19) and Kim (61)) are considered to be task times of Model-1 for the Operator. Afterward, the task times of other models are randomly derived from being within +- 10 % of these times. Since Thomopoulos (19) and Kim (61) are mixed-model benchmark problems, there is no need to produce additional data for other models. After the Operator data is determined, the task times of other resources (cobot types) are randomly derived such that Cobot Type-1 is 10 %-15 % shorter than Operator, and Cobot Type-2 is 5 %-10 % longer than Cobot Type-1. To be more realistic, the task times for all models of all resource types are set equal to zero with a 5 % probability. Precedence relationship diagrams in benchmark problems are used as common precedence relationship diagrams. The time-limit parameter applied to the Gurobi solver is 3600 seconds for MILP-1, MILP-2, and multi-objective, and the absolute and relative GAP is set to equal zero in all solutions.

The parameters of the benchmark problems are presented in Table 5, and the solutions obtained with the Gurobi solver, which is the exact solution method, are presented in Table 6. Please note that the “INF” value in the column “Objective Value” in Table 6 means no feasible solution is reached within the defined time-limit. Additionally, if GAP is not equal to zero for both MILP-1 and MILP-2, the multi-objective cannot be calculated for that benchmark problem since the min-max values used in the normalization calculation are not finalized. Afterward, the same benchmark problems are solved with NSGA-II, which is developed as an alternative to the exact

Table 2
Data for the cost and energy consumed by the cobots.

Type	Line	Workstation	Operator	Cobot Type-1	Cobot Type-2
Cost (Euro/ Time unit)	1.50	1.40	0.70	0.10	0.08
Energy Consumption During Operation ($PM_{2.5}$)	—	—	—	0.43	0.38
Energy Consumption During Standby ($PM_{2.5}$)	—	—	—	0.043	0.038
Average $PM_{2.5}$ emission coefficient of the energy consumption (g/kWh)	—	—	—	2.4956	

Table 3
Objective values of the optimal solutions.

Objective	Scenario-1		Scenario-2	
	1	2	1	2
$\boldsymbol{obj}_1$	**3240**	1637.40	**2960**	1651.89
$\boldsymbol{obj}_2$	4060	**366.07**	3628	**364.97**

Table 4
Optimal solutions for two different scenarios.

Data	Objective	Objective Value	Solution Time (sec)	Number of Lines Opened	Number of Opened Workstations (Line 1/ Line 2)	Models Assigned to Lines (Line 1/ Line 2)	Number of Operators Used	Number of Cobots Used (Type 1/Type 2)
Scenario-1	$\boldsymbol{obj}_1$	3240	6.47	2	4/4	1,2/3	2	2/5
	$\boldsymbol{obj}_2$	366.07	1.29	2	4/4	1,2/3	8	1/5
	$\boldsymbol{obj}_{\mathbf{Multi}}$	0.76 **(3520-907.51)**	4.41	2	4/4	1,2/3	4	2/5
Scenario-2	$\boldsymbol{obj}_1$	2960	89.16	2	3/4	1/2,3	2	2/5
	$\boldsymbol{obj}_2$	364.97	19.68	2	3/4	1/2,3	7	2/3
	$\boldsymbol{obj}_{\mathbf{Multi}}$	0.76 **(3224-871.81)**	46.80	2	3/4	1/2,3	4	2/4

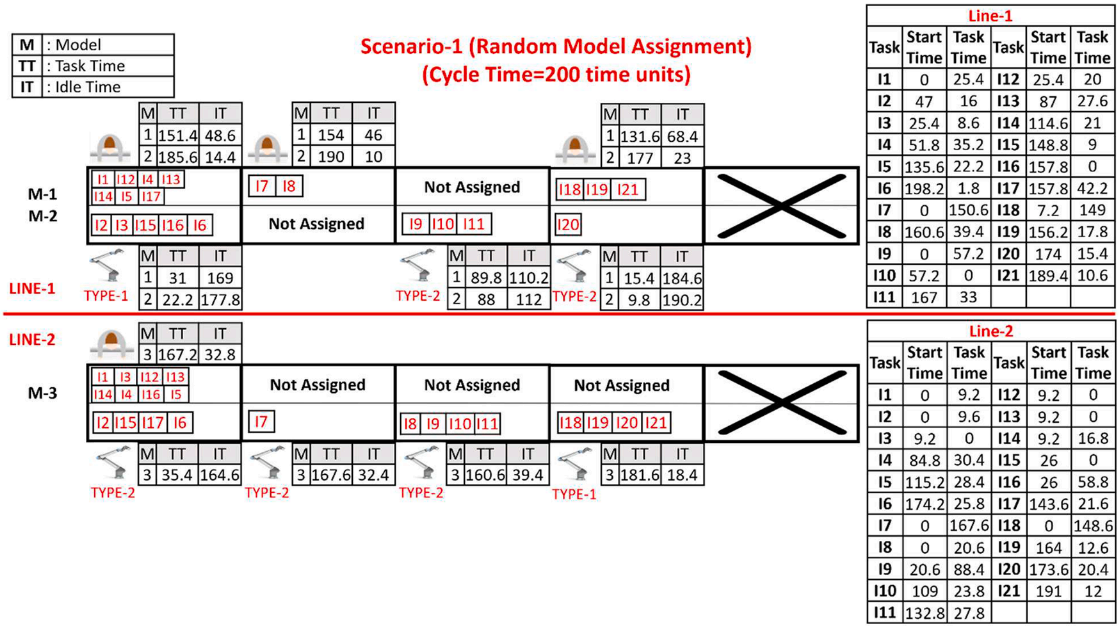

Fig. 4. Optimal balancing solution based on Scenario-1.

solution method.

When the results of the exact solution method regarding the benchmark problems in the literature are examined:

- The number of workstations used on each line in the solution of MILP-1 is used as the upper limit for MILP-2 and multi-objective function. Because of this rule, in the solution of MILP-2 for Jackson (11), the mathematical model is unable to open the third workstation on Line 1 to assign more tasks from cobots to operators. In the absence of this rule, MILP-2 opens all available workstations and tries to reduce the tasks assigned to cobots.
- The assignment of models to lines, the number of opened lines and workstations, and the number of assigned operators and cobots can vary depending on objective functions. While MILP-1 focuses on using as few resources as possible because it is cost-oriented, MILP-2 prioritizes operator usage. For this reason, while doing the same things, the number of resources used in each objective function's best solution can vary.

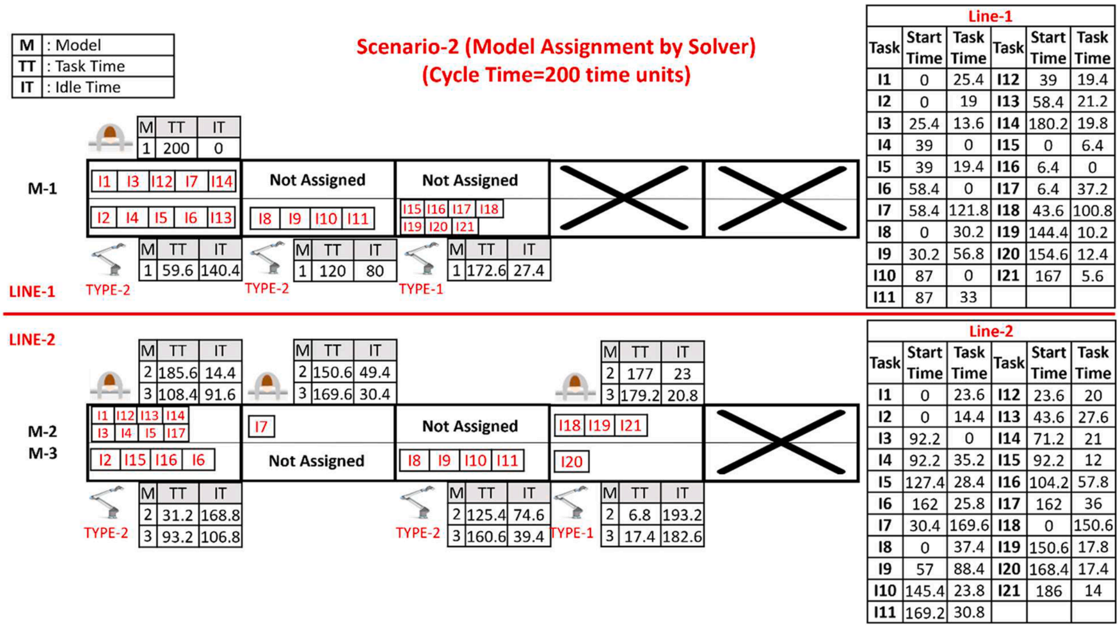

Fig. 5. Optimal balancing solution based on Scenario-2.

Table 5
Descriptive data for benchmark problems.

Benchmark Problem	Number of Tasks	Cycle Time (time units)	Number of W. Stations (max)	Number of Lines (max)	Number of Models and Demands	Planning Periods (time units)	Number of Operators	Number of Cobots (Type 1/ Type 2)
Bowman	8	25	2	2	3 (200/300/ 300)	17000	4	4/4
Jackson	11	10	3	2	3 (200/300/ 200)	6000	6	6/6
Thomopoulos	19	2	4	2	3 (200/300/ 200)	1200	8	8/8
Roszieg	25	23	4	2	4 (300/300/ 300/300)	16000	8	8/8
Heskia	28	150	5	2	4 (200/300/ 200/300)	100000	10	10/10
Lutz	32	2500	5	2	4 (200/400/ 300/300)	1900000	10	10/10
Gunther	35	75	5	2	4 (300/400/ 200/300)	57000	10	10/10
Kilbrid	45	65	5	2	4 (300/400/ 200/300)	50000	10	10/10
Hahn	53	2300	5	2	4 (300/400/ 300/200)	1700000	10	10/10
Warnecke	58	230	5	2	4 (200/200/ 200/300)	120000	10	10/10
Kim	61	18	5	2	4 (200/400/ 300/300)	12600	10	10/10
Tonge	70	500	5	2	4 (200/200/ 300/300)	310000	10	10/10
Arc	83	12000	6	2	4 (300/400/ 300/200)	8500000	12	12/12

- Among these benchmark problems in the time-limit (3600 seconds), the optimal solution (GAP=0) was achieved for all objective functions (obj_1, obj_2, and obj_{Multi}) only for Bowman (8), Jackson (11), Thomopoulos (19), and Roszieg (25).
- For Heskia (28), Lutz (32), Gunther (35), Kilbrid (45), and Hahn (53), there are solutions where GAP is not equal to zero in either or both MILP-1 and MILP-2. The multi-objective cannot be calculated for these benchmark problems since the min-max values used in normalization calculation are not finalized.
- For Warnecke (58) and later, no feasible solution can be found for all objectives in the defined time-limit (3600 seconds).

Table 6
Solutions obtained with Gurobi solver.

Benchmark Problem	Objective	Objective Value	Solution Time (sec)	GAP (%)	Number of Opened W.Stations (Line 1/ Line 2)	Models Assigned to Lines (Line 1/ Line 2)	Number of Operators Used	Number of Cobots Used (Type 1/ Type 2)
Bowman (8)	obj_1	258.50	0.35	0	2/2	2/1,3	2	1/3
	obj_2	91.79	0.36	0	2/2	1,2/3	4	2/2
	obj_{Multi}	0.633 **(259-117.87)**	0.44	0	2/2	1,2/3	2	2/2
Jackson (11)	obj_1	139.40	2.34	0	2/3	1/2,3	5	2/3
	obj_2	67.08	0.32	0	2/3	1/2,3	5	2/3
	obj_{Multi}	0 **(139.40-67.08)**	1.02	0	2/3	1/2,3	5	2/3
Thomopoulos (19)	obj_1	23.44	220.69	0	2/2	3/1,2	4	0/4
	obj_2	8.70	92.71	0	2/2	3/1,2	4	2/2
	obj_{Multi}	0.12 **(23.44-8.79)**	54.03	0	2/2	3/1,2	4	0/4
Roszieg (25)	obj_1	406.64	222.61	0	4/4	2,4/1,3	4	2/6
	obj_2	146.70	2249.34	0	4/4	1,4/2,3	8	0/8
	obj_{Multi}	0.89 **(438.38-206.28)**	978.24	0	4/4	2,4/1,3	6	1/7
Heskia (28)	obj_1	2856	3600	0.034	4/4	1,2/3,4	6	0/8
	obj_2	1223.14	3600	0.004	4/4	1,3/2,4	8	1/5
	obj_{Multi}	-	-	-	-	-	-	-
Lutz (32)	obj_1	47600	3600	0.083	4/4	1,2/3,4	6	0/8
	obj_2	15815.20	3600	0.044	4/4	1,3/2,4	8	2/5
	obj_{Multi}	-	-	-	-	-	-	-
Gunther (35)	obj_1	1380	3600	0.104	4/4	1,3/2,4	5	3/5
	obj_2	615.98	3600	0.032	4/4	1,3/2,4	8	1/5
	obj_{Multi}	-	-	-	-	-	-	-
Kilbrid (45)	obj_1	1526.20	3600	0.161	5/5	1,3/2,4	8	4/6
	obj_2	INF	3600	-	-	-	-	-
	obj_{Multi}	-	-	-	-	-	-	-
Hahn (53)	obj_1	INF	3600	-	-	-	-	-
	obj_2	16685.502	3600	0.306	5/5	1,4/2,3	10	2/8
	obj_{Multi}	-	-	-	-	-	-	-
Warnecke (58)	obj_1	INF	3600	-	-	-	-	-
	obj_2	INF	3600	-	-	-	-	-
	obj_{Multi}	-	-	-	-	-	-	-
Kim (61)	obj_1	INF	3600	-	-	-	-	-
	obj_2	INF	3600	-	-	-	-	-
	obj_{Multi}	-	-	-	-	-	-	-
Tonge (70)	obj_1	INF	3600	-	-	-	-	-
	obj_2	INF	3600	-	-	-	-	-
	obj_{Multi}	-	-	-	-	-	-	-
Arc (83)	obj_1	INF	3600	-	-	-	-	-
	obj_2	INF	3600	-	-	-	-	-
	obj_{Multi}	-	-	-	-	-	-	-

- The exact solution method can find the optimal solutions within acceptable times only for small-sized problems.
- The fact that the failure to achieve the optimal solutions within the defined time-limit begins with medium-sized benchmark problems is the most important indicator of the difficulty level of the problem.

4.3. NSGA-II – Meta-heuristic approach

As a result of not being able to reach the optimal solution for medium-sized problems due to the problem's difficulty, NSGA-II is developed as an alternative to the exact solution method. NSGA-II is one of the most popular and effective genetic algorithm-based algorithms in solving multi-objective problems. This algorithm is basically based on Pareto dominance. The most important steps are non-dominated sorting and crowding distance. The crowding distance operator prioritizes diversity when choosing among non-

Table 7
Combinations for Kilbrid (45).

Multi-Line Combination (*MLC*)	Line 1	Line 2
MLC 1	*C* 1: Model 1, Model 2	*C* 2: Model 3, Model 4
MLC 2	*C* 3: Model 1, Model 3	*C* 4: Model 2, Model 4
MLC 3	*C* 5: Model 1, Model 4	*C* 6: Model 2, Model 3

dominated solutions [36]. Since in this paper, the model-line assignment problem and the multiple assembly line balancing problem are solved together, the steps of the meta-heuristic algorithm are revised in accordance with this problem.

Firstly, model-line assignment combinations are determined according to the demands of the models and the planning period. To explain with the Kilbrid (45), planning period = 50000 and cycle time = 65. Thus, an assembly line can produce 50000/65 ≅ 769 products within the planning period. In this case, the total demand of the models assigned to a line cannot exceed 769. Please note that due to planning period constraints and model demands, for instance, Model 1, Model 2 and Model 3 cannot be assigned to the same line. There are three multi-line combinations (*MLCs*) and six model-line assignment combinations (*Cs*) for the Kilbrid (45); these combinations are presented in Table 7.

An initial population is created for each of these six *Cs*, and crossover, mutation, and NSGA-II operations are iteratively applied to these populations. In all benchmark problems, the size of the populations produced for each *Cs* is set to approximately 65 % of the number of tasks, and in all cases, it is between 10-60. At every n^{th} iteration, these six *Cs* are combined to create three *MLCs*. As presented in Table 7, *C* 1 containing Model 1 and Model 2 is combined with *C* 2 containing Model 3 and Model 4 to create *MLC* 1, *C* 3 containing Model 1 and Model 3 is combined with *C* 4 containing Model 2 and Model 4 to create *MLC* 2, and *C* 5 containing Model 1 and Model 4 is combined with *C* 6 containing Model 2 and Model 3 to create *MLC* 3. For each *MLC*, the number of solutions is the

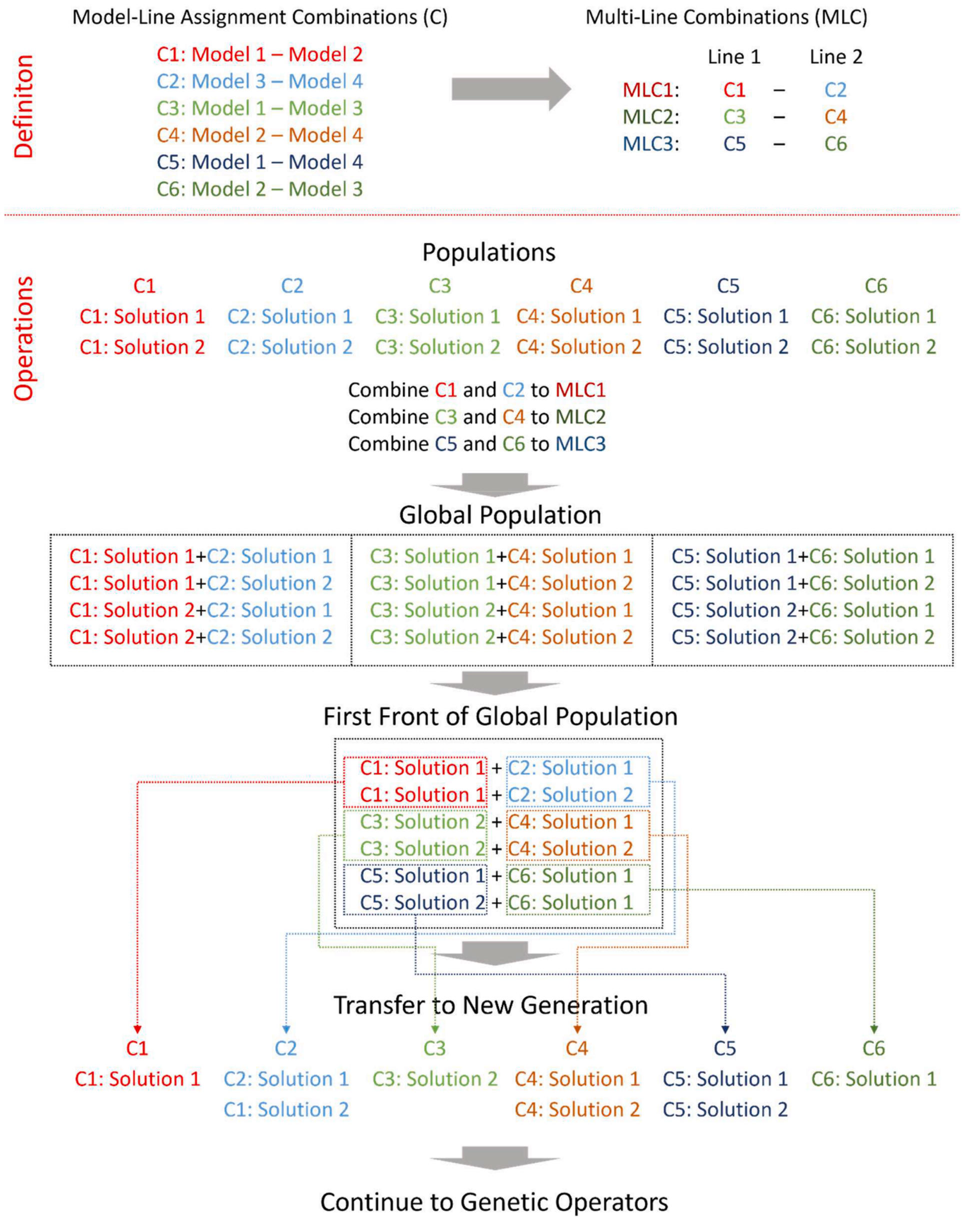

Fig. 6. The combination operations applied at each n^{th} iteration.

multiplication of the number of populations of the two merging *Cs*. The global population is acquired by the solutions obtained through creating these three *MLCs*. Only the first front (the first non-dominated solution list) of this global population is taken and transferred to the next generation. While being transferred to the next generation, the *Cs* that constitute this selected first front are equalized to their population. The iterative processes continue separately until the next n^{th} iteration, where the next combination (*Cs* to *MLCs*) will occur. The combination of operations applied at each n^{th} iteration is shown in Fig. 6. After each combination, since some poorly performing *Cs* will not be passed on to the next generation, as generations pass, only those *Cs* that create the best *MLC* will remain in the global population. The largest number of generations among the solutions that constitute the first front selected after each combination is considered as the iteration in which the last improvement occurs. The absence of improvement during the defined

```
Initialize parameters
Determine appropriate model-line assignment combinations (Cs) based on model demands and planning period
for i = 1: Cs.count
    Initialize Population Pc
    TaskAssignment(Population Pc)
    CalculateObjective(Population Pc)
end for
while (iter - last_improvement_iter) < not_improvement_limit
    iter++
    for i = 1: Cs.count
        Determine Population Qc
        Determine Population Rc
        for j = 1: Population Pc * crossover_rate
            Population Qc.add(Crossover(Population Pc))
        end for
        for j = 1: Population Pc * mutation_rate
            random_value=Random()
            if  random_value < 0.5
                Population Qc.add(InsertMutation(Population Pc))
            else
                Population Qc.add(SwapMutation(Population Pc))
            end if
        end for
        TaskAssignment(Population Qc)
        CalculateObjective(Population Qc)
        Population Rc = Population Pc ∪ Population Qc
        Determine temp_Population Pc
        while temp_Population Pc.count < population_size
            selected_list=SelectFront(Population Rc)
            if  (temp_Population Pc.count + selected_list.count) > population_size
                CrowdingDistance(selected_list)
            end if
            temp_Population Pc.add(selected_list)
            Population Rc.delete(selected_list)
        end while
        Population Pc = temp_Population Pc
    end for
    if  (iter % population_size) == 0
        Determine Global_Population
        for i = 1: MLCs.count
            Global_Population.add(CombineModelLine(Population Pc (Line 1), Population Pc (Line 2)))
        end for
        Global_Population = SelectFront(Global_Population)
        last_improvement_iter = Global_Population.maxiter
        for i = 1: Cs.count
            Population Pc = Global_Population.select(Pc)
        end for
    end if
end while
return Global_Population
```

Fig. 7. Pseudocode of NSGA-II.

number of iterations is accepted as the stopping criterion. After the stopping criterion, the first front of the current global population constitutes the best solution. The pseudocode of NSGA-II, which is described above, is shown in Fig. 7. The solution approach applies the single-point crossover operator with a ratio of 0.6. The mutation operator is applied with an equal probability of Insert Mutation or Swap Mutation, with a ratio of 0.1. Illustrations of genetic operators are shown in Fig. 8.

Benchmark problems where the performance of the mathematical model is tested with the exact solution method are solved with the presented NSGA-II procedure. Three runs are taken for each benchmark problem. The first front of these runs and the first front of the common solution list created by combining these three first fronts are taken. The first front of the common solution list is called "Super Pareto". In other words, "Super Pareto" represents the non-dominated solution set obtained by merging the first fronts of multiple independent runs, ensuring a more comprehensive approximation. This approach helps to mitigate the variability of heuristic algorithms by providing a broader perspective on the solution space [82]. The first front of the three runs and "Super Pareto" are shown in the graph created for each benchmark problem. Solutions within "Super Pareto" are connected by dashed lines. These graphics are shown in Fig. 9 and Fig. 10. Detailed information about the first fronts of the three runs taken for each benchmark problem is in the supplementary materials, and detailed information about the "Super Pareto" for each benchmark problem is in Appendix C. In addition, the convergence of the generations are shown in Fig. 11 for the objective function values over Kilbrid (45).

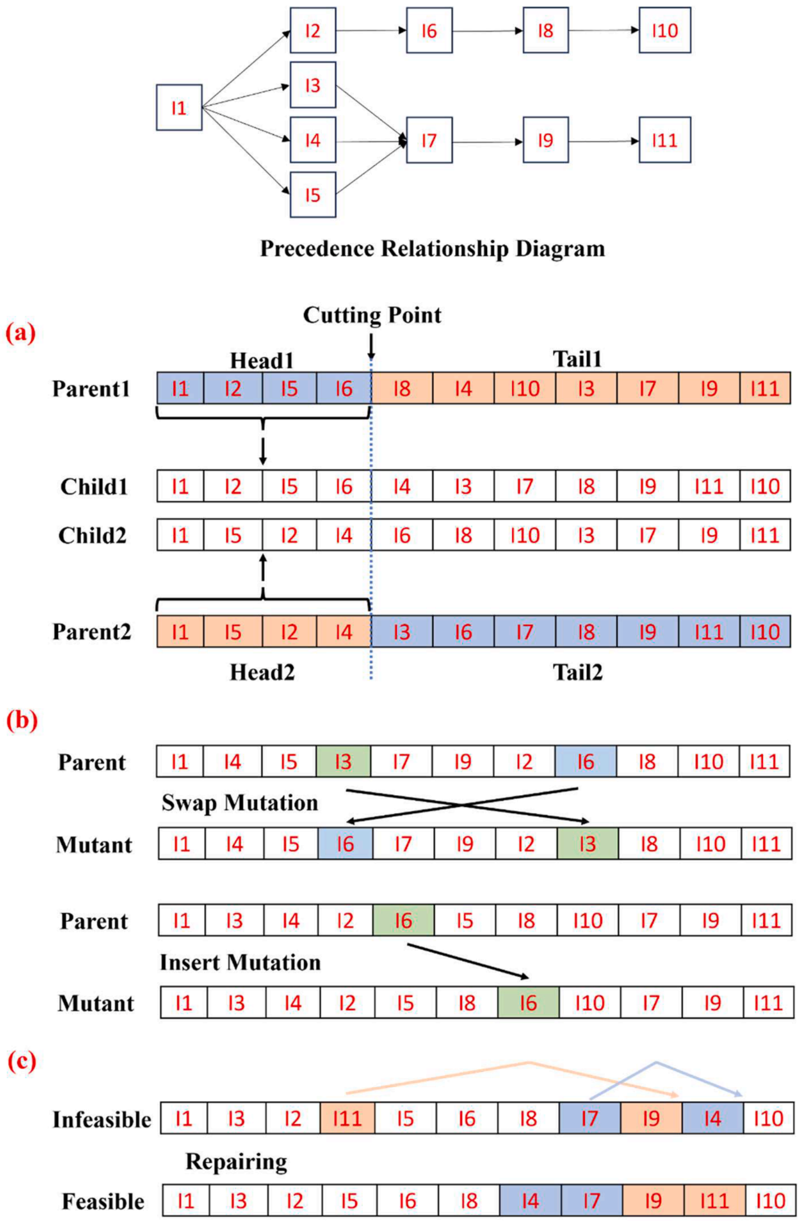

Fig. 8. Genetic Operators: (a) Crossover, (b) Mutation, (c) Repairing Mechanisms.

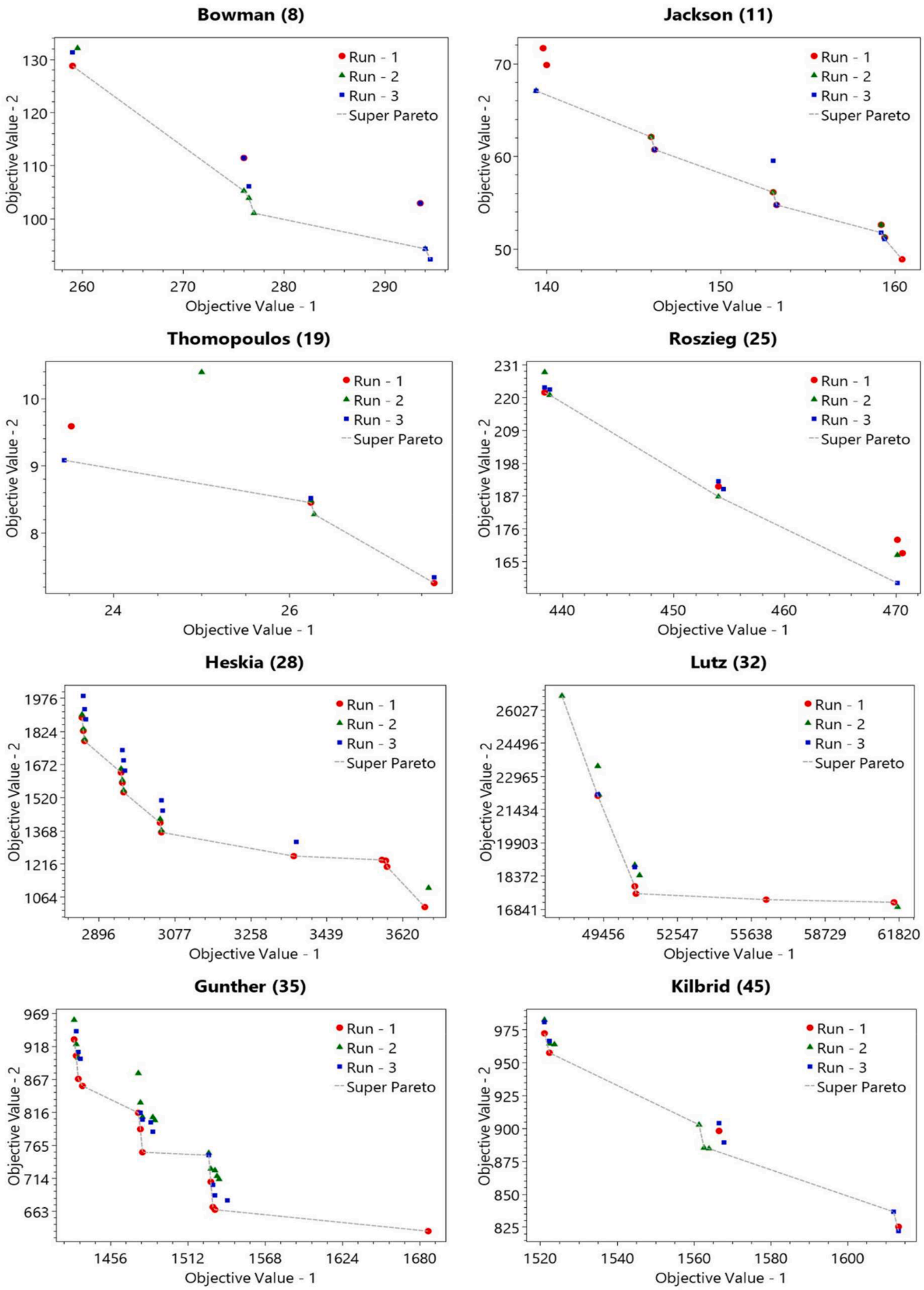

Fig. 9. Super pareto solutions of Bowman (8) – Kilbrid (45).

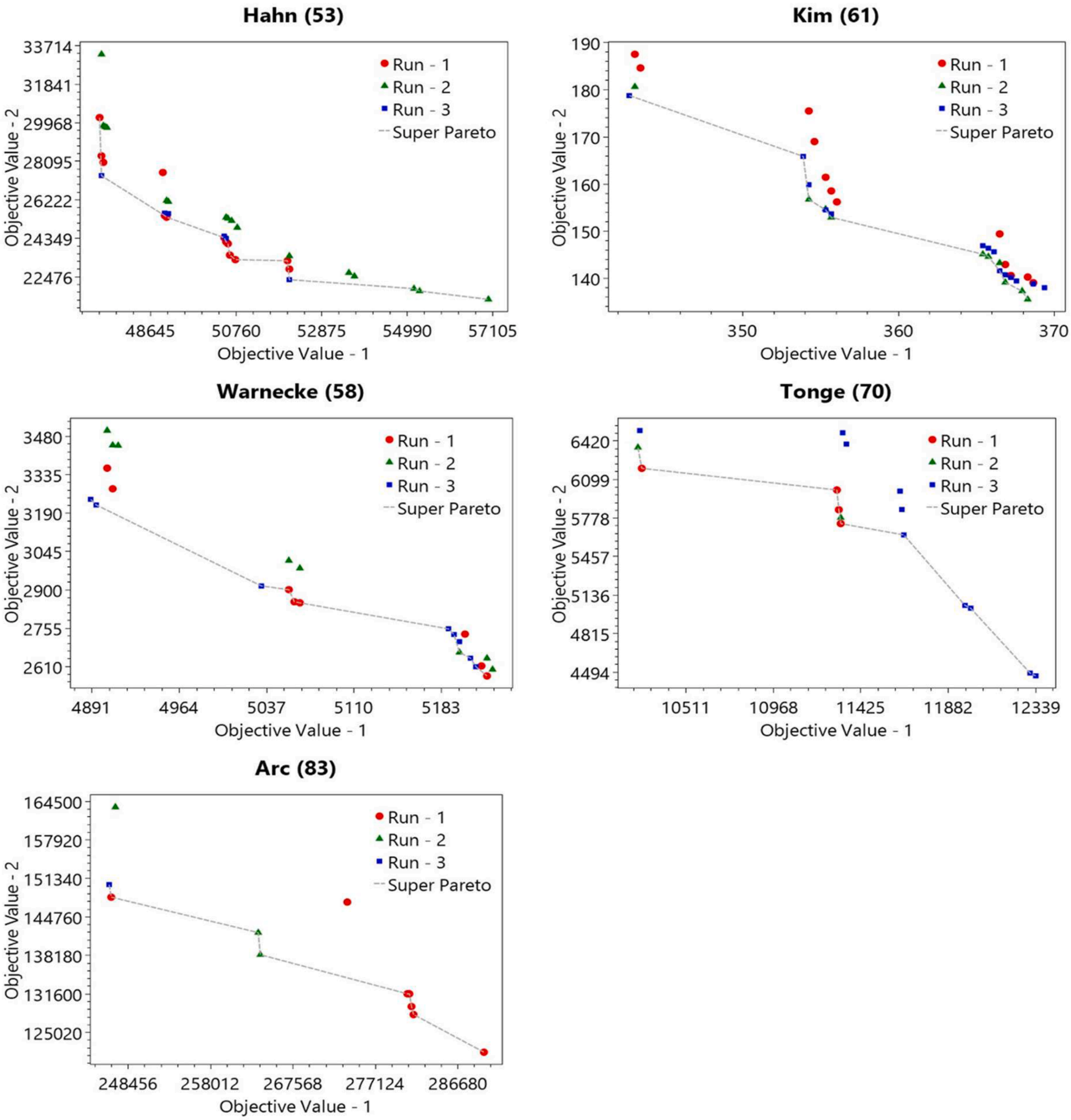

Fig. 10. Super Pareto solutions of Hahn (53) – Arc (83).

When the NSGA-II results of the benchmark problems that are used in the performance test of the exact solution method are examined:

- In benchmark problems where the exact solution method can reach the optimal result within the defined time-limit (Bowman (8), Jackson (11), Thomopoulos (19), Roszieg (25)) and in benchmark problems where the exact solution method can find a feasible result even if it cannot reach the optimal result (Heskia (28)), Lutz (32), Gunther (35), Kilbrid (45) Hahn (53)), NSGA-II can produce alternative solutions including optimal results in a much shorter time.
- NSGA-II can also reach quality solutions within acceptable times in benchmark problems (Warnecke (58) and later) where the exact solution method cannot find any results within the defined time-limit.
- Since MILP-2 focuses on minimizing the total amount of $PM_{2.5}$ emission arising from the energy consumed by cobots, it assigns tasks to operators as much as possible. This preference sensitivity makes achieving optimal results in MILP-2 more difficult than MILP-1.

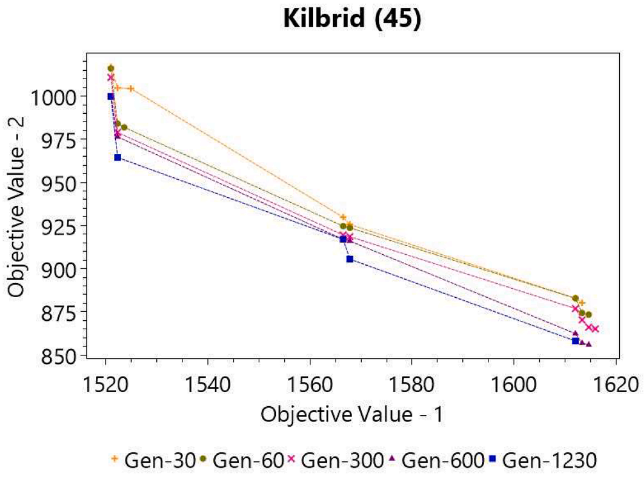

Fig. 11. Convergence of the generations for Kilbrid (45).

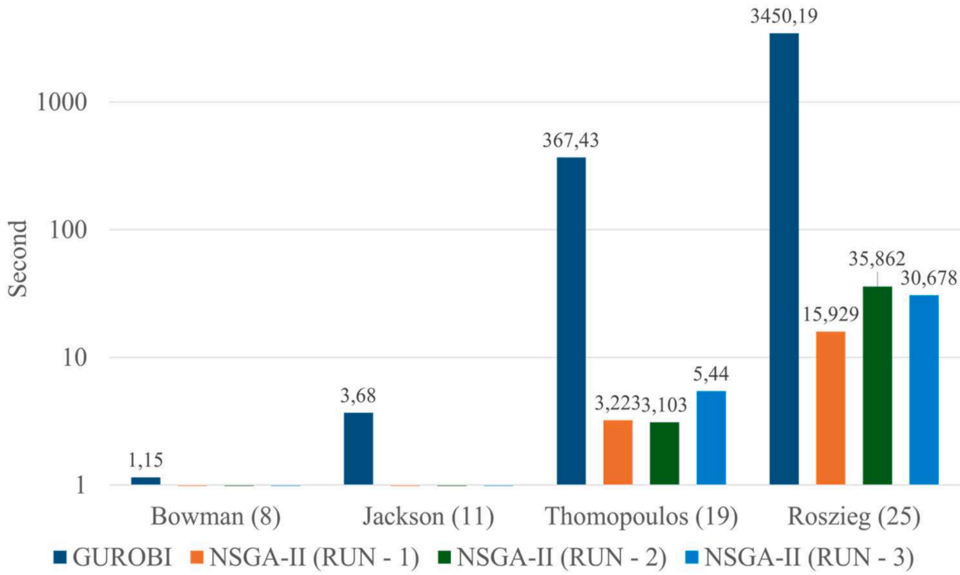

Fig. 12. Exact solution method (Gurobi) and NSGA-II solution time comparison.

For this reason, the quality of the MILP-2 value of the results found by NSGA-II, especially for medium and large-sized problems, is 92 % and above.

- NSGA-II can produce results much faster than the exact solution method on all benchmark problems. Fig. 12 shows the solution times for the four benchmark problems in which the exact solution method can reach the optimal result and the times of the three runs of NSGA-II taken for each benchmark problem.

The overall findings in this research highlight the practical advantages of integrating robots into assembly lines across various industries. In practice, robotic utilization can lead to cost reductions, improved energy efficiency, and increased production speed. Additionally, robots enhance operational flexibility in complex manufacturing environments, allowing manufacturers to better manage production variability and optimize resource allocation.

NSGA-II delivers solutions comparable to exact methods for small- and medium-scale problems, achieving optimal results in tractable computation times. For large-scale industrial problems where exact methods become computationally prohibitive, the proposed approach demonstrates robust scalability, generating high-quality solutions within practical time limits. While absolute optimality cannot be formally guaranteed for these larger instances, the method's strong performance on smaller problems (considering stable convergence behavior across varying problem scales) provides evidence of solution quality. Notably, the algorithm exhibits a characteristic and manageable trade-off between solution quality and computation time that aligns with expectations for NP-hard optimization problems.

5. Conclusions

This study proposes a simultaneous model-line assignment and robotic mixed-model multiple assembly line balancing (MLA-RMMALB) problem. In the multi-objective mathematical model, (i) production cost and (ii) total $PM_{2.5}$ emission arising from the energy consumed by cobots are optimized simultaneously. In the problem, mixed-model production is allowed on each line. Cobots are heterogeneous; not all types can perform every task, and some may require different processing times for the same task. One operator and one cobot can work at each workstation in parallel, and the cycle time is fixed.

The integration of model-line assignment within the mixed-model robotic multiple assembly line balancing problem addresses a notable gap in the literature. In addition, there is a lack of literature on studies that simultaneously examine the objectives, such as production cost and energy consumption of cobots.

The proposed mathematical model is tested with two scenarios with different model-line assignment strategies on a numerical example containing 21 tasks. In scenario-1, models are randomly assigned to the lines, and in scenario-2, models are assigned to the lines. This numerical example is solved separately for each objective function and subsequently for the combined multi-objective formulation to examine the effects of the objective functions on multiple assembly lines. The results indicate that the mathematical model generates high-quality solutions aligned with the intended objectives. In addition, it is seen that the model-line assignment process performed by the solution approach significantly improves the results. Still, the solution time increases due to the growth of the feasible region.

To test the performance of the model, the benchmark problems (Bowman (8), Jackson (11), Thomopoulos (19), Roszieg (25), Heskia (28), Lutz (32), Gunther (35), Kilbrid (45), Hahn (53), Warnecke (58), Kim (61), Tonge (70), and Arc (83)) in literature, whose number of tasks gradually increases, are adapted in accordance with the problem to produce three or four models and solved. According to the results of the performance test, while the mathematical model can achieve quality results in small-sized samples, there are difficulties in finding the optimal solution within acceptable times. In some solutions, no results can be achieved in medium and large-sized samples.

Simultaneous solving of model-line assignment and robotic mixed-model multiple assembly line balancing problem, cobots being heterogeneous, each resource not being able to do every task (capability matrix) or doing it at different times, an operator and a cobot being able to work simultaneously at the same workstation (scheduling), optimizing multiple objectives at the same time are the most important factors that make the problem difficult.

To obtain quality solutions within these difficulties and acceptable times, genetic algorithm-based NSGA-II, which is popular and effective in multi-objective problems in the literature, is developed as an alternative to the exact solution method. Since in this paper, the model-line assignment problem and the multiple assembly line balancing problem are solved together, the steps of the meta-heuristic algorithm are adapted to address the specific characteristics of this problem. NSGA-II is applied to the same benchmark problems used in evaluating the exact solution method. When the results are examined, it is observed that NSGA-II can produce alternative solutions, including optimal solutions, in a very short time for small and medium-sized problems and quality solutions within acceptable times for large-sized problems. In addition, since MILP-2 focuses on assigning tasks to operators as much as possible, it can reach a quality of 92 % and above, especially in medium and large-sized problems.

The Super Pareto solutions generated in this study establish a comprehensive benchmark that enables direct performance comparisons across various metaheuristic approaches. These detailed solution sets provide a valuable foundation for future comparative studies examining alternative optimization methodologies (even including game theoretical approaches, e.g. Stackelberg model [83]).

Despite the strengths of the proposed approach, three limitations should be noted. First, as the number of objective functions increases, NSGA-II encounters greater difficulty in identifying optimal solutions due to expanding solution space complexity. Second, for medium- and large-sized problems, while NSGA-II maintains high-quality solutions for primary objectives, secondary objective performance shows measurable degradation as problem size increases. Third, solution times frequently exceed practical thresholds in large-scale implementations, challenging computational efficiency.

From a practical perspective, these findings demonstrate that while our model and NSGA-II implementation provide substantial advances in mixed-model multi-assembly line balancing, additional refinements remain necessary to boost computational efficiency - particularly for large-scale industrial implementations.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cam.2025.116876.

Appendix A. Task times for resources

Task	Operator Model (1/2/3)			Cobot Type-1 Model (1/2/3)			Cobot Type-2 Model (1/2/3)		
1	25.4	23.6	9.2	20.4	18.6	4.2	23.4	21.6	7.2
2	21	16.4	11.6	16	11.4	6.6	19	14.4	9.6

(continued on next page)

(*continued*)

Task	Operator Model (1/2/3)			Cobot Type-1 Model (1/2/3)			Cobot Type-2 Model (1/2/3)		
3	13.6	0	0	8.6	0	0	11.6	0	0
4	0	35.2	30.4	0	30.2	25.4	0	34.2	28.4
5	21.4	22.2	28.4	16.4	17.2	23.4	19.4	20.2	26.4
6	0	6.8	27.8	0	1.8	22.8	0	4.8	25.8
7	121.8	150.6	169.6	116.8	145.6	164.6	119.8	148.6	167.6
8	32.2	39.4	22.6	27.2	34.4	17.6	30.2	37.4	20.6
9	58.8	59.2	90.4	53.8	54.2	85.4	56.8	57.2	88.4
10	0	0	25.8	0	0	20.8	0	0	23.8
11	35	32.8	30.8	30	27.8	25.8	33	30.8	27.8
12	19.4	20	0	14.4	15	0	17.4	18	0
13	23.2	27.6	0	18.2	22.6	0	21.2	25.6	0
14	19.8	21	16.8	14.8	16	11.8	17.8	19	14.8
15	11.4	14	0	6.4	9	0	9.4	12	0
16	0	0	58.8	0	0	53.8	0	0	57.8
17	42.2	36	23.6	37.2	31	18.6	40.2	34	21.6
18	105.8	149	150.6	100.8	144	145.6	103.8	147	148.6
19	15.2	17.8	14.6	10.2	12.8	9.6	13.2	15.8	12.6
20	17.4	11.8	22.4	12.4	6.8	17.4	15.4	9.8	20.4
21	10.6	10.2	14	5.6	5.2	9	8.6	8.2	12

Appendix B. Capability matrix for resource types

Resource Type	Task																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Operator	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	0	1
Cobot Type-1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
Cobot Type-2	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0

Appendix C. Super Pareto solutions of benchmark problems obtained with NSGA-II

Benchmark Problem	Average Solution Time (sec)	obj_1	obj_2	Number of Opened W. Stations (Line 1/ Line 2)	Models Assigned to Lines (Line 1/ Line 2)	Number of Operators Used	Number of Cobots Used (Type 1/ Type 2)
Bowman (8)	0.08	259	128.80	2/2	1/0,2	2	2/2
		294	94.35	2/2	0/1,2	4	2/2
		276	105.27	2/2	1/0,2	3	1/3
		276.50	103.90	2/2	1/0,2	3	2/2
		277	101.07	2/2	2/0,1	3	3/1
		294.50	92.37	2/2	2/0,1	4	3/1
Jackson (11)	0.37	153.20	54.77	3/3	1/0,2	5	1/4
		153	56.14	3/3	1/0,2	5	0/5
		146.20	60.74	3/3	1/0,2	4	1/4
		146	62.11	3/3	1/0,2	4	0/5
		160.40	48.90	3/3	2/0,1	6	2/3
		139.40	67.09	2/3	0/1,2	5	2/3
		159.40	51.09	3/3	2/0,1	6	1/3
		159.20	51.77	3/3	2/0,1	6	0/4
Thomopoulos (19)	3.92	27.64	7.25	2/3	0/1,2	5	0/4
		26.24	8.45	2/3	0/1,2	4	0/4
		26.28	8.27	2/3	0/1,2	4	1/3
		23.44	9.08	2/2	2/0,1	4	0/4
Roszieg (25)	27.48	438.38	221.78	4/4	0,2/1,3	6	1/7
		438.84	220.94	4/4	0,2/1,3	6	2/6
		454.02	186.82	4/4	0,2/1,3	7	0/8
		470.12	157.80	4/4	0,2/1,3	8	0/8
Heskia (28)	55.50	3360	1252.81	5/4	0,1/2,3	9	1/5
		3042	1405.70	4/4	0,1/2,3	8	0/6
		2952	1589.53	4/4	0,1/2,3	7	1/6
		2949	1636.43	4/4	0,1/2,3	7	0/7
		2862	1780.81	4/4	0,1/2,3	6	2/6
		2859	1827.71	4/4	0,1/2,3	6	1/7
		3045	1361.99	4/4	0,1/2,3	8	1/5

(*continued on next page*)

(*continued*)

Benchmark Problem	Average Solution Time (sec)	obj_1	obj_2	Number of Opened W. Stations (Line 1/ Line 2)	Models Assigned to Lines (Line 1/ Line 2)	Number of Operators Used	Number of Cobots Used (Type 1/ Type 2)
		2955	1545.81	4/4	0,1/2,3	7	2/5
		2856	1888.50	4/4	0,1/2,3	6	0/8
		3672	1019.07	5/5	0,3/1,2	10	0/6
		3582	1203.86	5/5	0,3/1,2	9	1/6
		3579	1231.87	5/5	0,3/1,2	9	0/7
		3570	1235.24	5/5	0,3/1,2	9	1/5
Lutz (32)	92.35	50750	17902.14	4/4	0,1/2,3	8	1/5
		49200	22079.62	4/4	0,1/2,3	7	1/6
		50800	17561.25	4/4	0,1/2,3	8	2/4
		56250	17285.06	4/5	0,1/2,3	9	2/5
		61600	17162.21	5/5	0,3/1,2	10	4/3
		47700	26687.62	4/4	0,1/2,3	6	2/6
		61750	16944	5/5	0,2/1,3	10	3/5
Gunther (35)	199.61	1435.50	857.11	4/4	0,1/2,3	6	5/3
		1432.50	867.90	4/4	0,1/2,3	6	3/5
		1429.50	928.93	4/4	0,1/2,3	6	1/7
		1431	903.67	4/4	0,1/2,3	6	2/6
		1479	754.49	4/4	0,1/2,3	7	3/4
		1476	815.52	4/4	0,1/2,3	7	1/6
		1477.50	790.26	4/4	0,1/2,3	7	2/5
		1531.50	665.80	4/4	0,1/2,3	8	3/4
		1530	669.88	4/4	0,1/2,3	8	2/5
		1528.50	708.84	4/4	0,1/2,3	8	1/6
		1686	632.61	4/5	0,2/1,3	9	5/1
		1527	749.98	4/4	0,1/2,3	8	4/2
Kilbrid (45)	931.76	1522.30	957.69	5/5	0,1/2,3	8	1/9
		1521	972.51	5/5	0,1/2,3	8	0/10
		1562.60	885.41	5/5	0,1/2,3	9	1/8
		1561.30	902.99	5/5	0,1/2,3	9	0/9
		1563.90	884.83	5/5	0,1/2,3	9	2/7
		1612	836.80	5/5	0,1/2,3	10	0/10
		1613.30	822.07	5/5	0,1/2,3	10	1/9
Hahn (53)	1124.99	50554	24095.71	5/4	0,3/1,2	8	3/6
		52026	23266.15	5/4	0,3/1,2	9	4/4
		47380	30229.37	5/4	0,3/1,2	6	4/5
		50600	23540.66	5/4	0,3/1,2	8	4/5
		48990	25461.38	5/4	0,3/1,2	7	4/5
		50738	23319.31	5/4	0,3/1,2	8	7/2
		50508	24180.91	5/4	0,3/1,2	8	6/2
		50462	24386.73	5/4	0,3/1,2	8	5/3
		49036	25383.66	5/4	0,3/1,2	7	5/4
		55292	21789.84	5/5	0,1/2,3	9	1/8
		55154	21907.38	5/5	0,1/2,3	9	2/6
		56994	21388.02	5/5	0,1/2,3	10	3/6
		47426	27400.96	5/4	0,3/1,2	6	5/4
		52072	22345.86	5/4	0,3/1,2	9	5/3
Warnecke (58)	1105.66	5060	2855.17	4/5	0,1/2,3	8	4/5
		5221	2574.86	4/5	0,1/2,3	9	4/5
		5064.60	2851.23	4/5	0,1/2,3	8	5/4
		5055.40	2901.11	4/5	0,1/2,3	8	3/6
		5198	2663.81	4/5	0,1/2,3	9	3/5
		5211.80	2609.69	4/5	0,1/2,3	9	2/7
		5193.40	2731.85	4/5	0,1/2,3	9	2/6
		5032.40	2915.26	4/5	0,1/2,3	8	2/6
		5207.20	2641.98	4/5	0,1/2,3	9	1/8
		4894.40	3221.62	4/5	0,1/2,3	7	3/6
		4889.80	3242.81	4/5	0,1/2,3	7	2/7
		5188.80	2752.84	4/5	0,1/2,3	9	1/7
Kim (61)	1813.75	354.24	156.75	4/4	0,1/2,3	7	1/6
		355.68	152.90	4/4	0,1/2,3	7	1/7
		366.84	139.16	4/4	0,1/2,3	8	1/6
		368.28	135.55	4/4	0,1/2,3	8	1/7
		367.92	137.31	4/4	0,1/2,3	8	0/8
		365.40	145.13	4/4	0,1/2,3	8	1/5
		365.76	144.61	4/4	0,1/2,3	8	2/4
		366.48	141.61	4/4	0,1/2,3	8	0/7
		353.88	165.88	4/4	0,1/2,3	7	0/7
		355.32	154.52	4/4	0,1/2,3	7	0/8

(*continued on next page*)

(*continued*)

Benchmark Problem	Average Solution Time (sec)	obj_1	obj_2	Number of Opened W. Stations (Line 1/ Line 2)	Models Assigned to Lines (Line 1/ Line 2)	Number of Operators Used	Number of Cobots Used (Type 1/ Type 2)
		342.72	178.80	4/4	0,1/2,3	6	0/8
Tonge (70)	2213.72	10280	6192.42	4/4	0,1/2,3	8	6/2
		11320	5732.28	5/4	0,3/1,2	9	5/3
		11300	6012.25	5/4	0,3/1,2	9	3/5
		11310	5847.65	5/4	0,3/1,2	9	4/4
		10260	6366.53	4/4	0,1/2,3	8	4/4
		12000	5026.91	5/5	0,2/1,3	9	3/5
		12340	4463.66	5/5	0,2/1,3	10	2/6
		11970	5050.39	5/5	0,2/1,3	9	4/3
		12310	4487.14	5/5	0,2/1,3	10	3/4
		11650	5638.19	5/5	0,2/1,3	8	3/5
Arc (83)	2190.11	246480	148164.90	4/4	0,1/2,3	8	5/3
		281280	129478.20	5/5	0,3/1,2	8	2/8
		281520	128105.80	5/5	0,3/1,2	8	3/7
		281040	131643	5/5	0,3/1,2	8	1/9
		280800	131652.20	5/5	0,3/1,2	8	0/10
		289680	121669.90	5/5	0,3/1,2	8	2/8
		263760	138324.60	5/4	0,1/2,3	8	3/6
		263520	142102	5/4	0,1/2,3	8	2/7
		246240	150296	4/4	0,1/2,3	8	4/4

Data availability

We have shared the link to the data within the manuscript.

References

[1] I. Kucukkoc, D.Z. Zhang, Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines, Int. J. Adv. Manuf. Technol. 82 (2016) 265–285.
[2] U. Ozcan, H. Gokcen, B. Toklu, Balancing parallel two-sided assembly lines, Int. J. Prod. Res. 48 (16) (2010) 4767–4784.
[3] M.-A. Abdous, X. Delorme, D. Battini, F. Sgarbossa, S. Berger-Douce, Assembly line balancing problem with ergonomics: a new fatigue and recovery model, Int. J. Prod. Res. 61 (3) (2023) 693–706.
[4] M.E. Salveson, The assembly line balancing problem, J. Ind. Eng. 6 (3) (1955) 18–25.
[5] K. Kara, Y. Atasagun, H. Gökçen, S. Hezer, N. Demirel, An integrated model to incorporate ergonomics and resource restrictions into assembly line balancing, Int. J. Comput. Integr. Manuf. 27 (11) (2014) 997–1007.
[6] H. Aguilar, A. García-Villoria, R. Pastor, A survey of the parallel assembly lines balancing problem, Comput. Oper. Res. 124 (2020) 105061.
[7] K. Meng, Q. Tang, Z. Zhang, Balancing and sequencing of mixed-model assembly line considering preventive maintenance scenarios: mathematical model and a migrating birds optimization algorithm, Flex. Serv. Manuf. J. 35 (4) (2023) 1175–1205.
[8] O. Battaïa, A. Dolgui, A taxonomy of line balancing problems and their solutionapproaches, Int. J. Prod. Econ. 142 (2) (2013) 259–277.
[9] Z. Li, I. Kucukkoc, Q. Tang, Z. Zhang, Models and two-phase bee algorithms for multi-objective U-shaped disassembly line balancing problem, Optim. Eng. 24 (1) (2023) 591–622.
[10] H. Soysal Kurt, S.K. Isleyen, H. Gokcen, Balancing and sequencing of mixed-model parallel robotic assembly lines considering energy consumption, Flex. Serv. Manuf. J. (2024) 1–29.
[11] M.C. Sahin, M.K. Tural, Robotic stochastic assembly line balancing, Flex. Serv. Manuf. J. 35 (4) (2023) 1076–1115.
[12] I. Kucukkoc, D.Z. Zhang, Type-E parallel two-sided assembly line balancing problem: Mathematical model and ant colony optimisation based approach with optimised parameters, Comput. Ind. Eng. 84 (2015) 56–69.
[13] N. Boysen, P. Schulze, A. Scholl, Assembly line balancing: What happened in the last fifteen years? Eur. J. Oper. Res. 301 (3) (2022) 797–814.
[14] P. Chutima, Assembly line balancing with cobots: An extensive review and critiques, Int. J. Ind. Eng. Comput. 14 (4) (2023) 785–804.
[15] M. Kheirabadi, S. Keivanpour, Y. Chinniah, J.-M. Frayret, A review on collaborative robot assembly line balancing problems, IFAC-Pap. 55 (10) (2022) 2779–2784.
[16] A. Lusa, A survey of the literature on the multiple or parallel assembly line balancing problem, Eur. J. Ind. Eng. 2 (1) (2008) 50–72.
[17] A. Otto, O. Battaïa, Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: A survey, Comput. Ind. Eng. 111 (2017) 467–480.
[18] P. Sivasankaran, P. Shahabudeen, Literature review of assembly line balancing problems, Int. J. Adv. Manuf. Technol. 73 (2014) 1665–1694.
[19] S.O. Tasan, S. Tunali, A review of the current applications of genetic algorithms in assembly line balancing, J. Intell. Manuf. 19 (2008) 49–69.
[20] A. Keshvarparast, N. Katiraee, A. Pirayesh, O. Battaia, N. Berti, Integrated Resource Optimization in a Multi-Product Separated Line Collaborative Assembly Line Balancing Problem (MPSLC-ALBP), IFAC-Pap. 56 (2) (2023) 713–718.
[21] Z. Mao, J. Zhang, K. Fang, D. Huang, Y. Sun, Balancing U-type assembly lines with human–robot collaboration, Comput. Oper. Res. 159 (2023) 106359.
[22] A. Nourmohammadi, M. Fathi, A.H. Ng, Balancing and scheduling assembly lines with human-robot collaboration tasks, Comput. Oper. Res. 140 (2022) 105674.
[23] P.T. Zacharia, E.K. Xidias, A.C. Nearchou, The fuzzy human-robot collaboration assembly line balancing problem, Comput. Ind. Eng. 187 (2024) 109774.
[24] G. Levitin, J. Rubinovitz, B. Shnits, A genetic algorithm for robotic assembly line balancing, Eur. J. Oper. Res. 168 (3) (2006) 811–825.
[25] J. Gao, L. Sun, L. Wang, M. Gen, An efficient approach for type II robotic assembly line balancing problems, Comput. Ind. Eng. 56 (3) (2009) 1065–1080.
[26] C. Weckenborg, T.S. Spengler, Assembly line balancing with collaborative robots under consideration of ergonomics: a cost-oriented approach, IFAC-Pap. 52 (13) (2019) 1860–1865.
[27] C. Weckenborg, K. Kieckhäfer, C. Müller, M. Grunewald, T.S. Spengler, Balancing of assembly lines with collaborative robots, Bus. Res. 13 (1) (2020) 93–132.
[28] C.G.S. Sikora, C. Weckenborg, Balancing of assembly lines with collaborative robots: comparing approaches of the Benders' decomposition algorithm, Int. J. Prod. Res. 61 (15) (2023) 5117–5133.

[29] Z.A. Cil, Z. Li, S. Mete, E. Ozceylan, Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration, Appl. Soft Comput. 93 (2020) 106394.
[30] G. Boschetti, M. Faccio, M. Milanese, R. Minto, C-ALB (Collaborative Assembly Line Balancing): a new approach in cobot solutions, Int. J. Adv. Manuf. Technol. 116 (2021) 3027–3042.
[31] M.F.F. Ab Rashid, N.M.Z.N. Mohamed, A.N Oumer, Modelling and optimization of energy efficient assembly line balancing using modified moth flame optimizer, Int. J. Integr. Eng. 14 (1) (2022) 25–39.
[32] L. Belkharroubi, K. Yahyaoui, Solving the energy-efficient robotic mixed-model assembly line balancing problem using a memory-based Cuckoo Search Algorithm, Eng. Appl. Artif. Intell. 114 (2022) 105112.
[33] T. Chitsrisakda, P. Chutima, A. Jiamsanguanwong, O. Kittithreerapronchai, Application of the NSGA-III algorithm to investigate the Cobot assembly line balancing problem with disabled workers, in: Proceedings of the 2024 6th International Conference on Management Science and Industrial Engineering, 2024.
[34] P. Chutima, A. Khotsaenlee, Multi-objective parallel adjacent U-shaped assembly line balancing collaborated by robots and normal and disabled workers, Comput. Oper. Res. 143 (2022) 105775.
[35] Y. Huang, B. Sheng, R. Luo, Y. Lu, G. Fu, X. Yin, Solving human-robot collaborative mixed-model two-sided assembly line balancing using multi-objective discrete artificial bee colony algorithm, Comput. Ind. Eng. 187 (2024) 109776.
[36] H. Babazadeh, M. Alavidoost, M.F. Zarandi, S. Sayyari, An enhanced NSGA-II algorithm for fuzzy bi-objective assembly line balancing problems, Comput. Ind. Eng. 123 (2018) 189–208.
[37] H. Babazadeh, N. Javadian, A novel meta-heuristic approach to solve fuzzy multi-objective straight and U-shaped assembly line balancing problems, Soft Comput. 23 (17) (2019) 8217–8245.
[38] Z. Li, M. Janardhanan, Q. Tang, Z. Zhang, Models and algorithms for U-shaped assembly line balancing problem with collaborative robots, Soft Comput. 27 (14) (2023) 9639–9659.
[39] W. Helgeson, D.P. Birnie, Assembly line balancing using the ranked positional weight technique, J. Ind. Eng. 12 (6) (1961) 394–398.
[40] I. Baybars, An efficient heuristic method for the simple assembly line balancing problem, Int. J. Prod. Res. 24 (1) (1986) 149–166.
[41] A.L. Arcus, A computer method of sequencing operations for assembly lines, Int. J. Prod. Res. 4 (4) (1965) 259–277.
[42] S. Avikal, R. Jain, P. Mishra, H. Yadav, A heuristic approach for U-shaped assembly line balancing to improve labor productivity, Comput. Ind. Eng. 64 (4) (2013) 895–901.
[43] D.A. Ajenblit, R.L. Wainwright, Applying genetic algorithms to the U-shaped assembly line balancing problem, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), 1998.
[44] E. Falkenauer, A. Delchambre, A genetic algorithm for bin packing and line balancing, ICRA (1992).
[45] F. Pistolesi, B. Lazzerini, M. Dalle Mura, G. Dini, EMOGA: a hybrid genetic algorithm with extremal optimization core for multiobjective disassembly line balancing, IEEE Trans. Ind. Inform. 14 (3) (2017) 1089–1098.
[46] Z. Zhang, Q. Tang, M. Chica, Z. Li, Reinforcement learning-based multiobjective evolutionary algorithm for mixed-model multimanned assembly line balancing under uncertain demand, IEEE Trans. Cybern. (2023).
[47] A. Baykasoglu, Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems, J. Intell. Manuf. 17 (2006) 217–232.
[48] N. Manavizadeh, N.-s. Hosseini, M. Rabbani, F. Jolai, A Simulated Annealing algorithm for a mixed model assembly U-line balancing type-I problem considering human efficiency and Just-In-Time approach, Comput. Ind. Eng. 64 (2) (2013) 669–685.
[49] S.D. Lapierre, A. Ruiz, P. Soriano, Balancing assembly lines with tabu search, Eur. J. Oper. Res. 168 (3) (2006) 826–837.
[50] C. Peterson, A tabu search procedure for the simple assembly line balancing problem, Proc. Decis. Sci. Inst. Conf. (1993).
[51] K. Wang, X. Li, L. Gao, P. Li, J.W. Sutherland, A discrete artificial bee colony algorithm for multiobjective disassembly line balancing of end-of-life products, IEEE Trans. Cybern. 52 (8) (2021) 7415–7426.
[52] N. Aydin, O. Yilmaz, M. Deveci, Z. Lv, Heuristics based optimization for multidepot drone location and routing problem to detect post-earthquake damages, IEEE Trans. Intell. Transp. Syst. 25 (1) (2022) 850–858.
[53] I. Sabuncuoglu, E. Erel, A. Alp, Ant colony optimization for the single model U-type assembly line balancing problem, Int. J. Prod. Econ. 120 (2) (2009) 287–300.
[54] J. Zha, J.-j. Yu, A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment, J. Manuf. Syst. 33 (1) (2014) 93–102.
[55] Y. Li, D. Liu, I. Kucukkoc, Mixed-model assembly line balancing problem considering learning effect and uncertain demand, J. Comput. Appl. Math. 422 (2023) 114823.
[56] M.N. Janardhanan, Z. Li, G. Bocewicz, Z. Banaszak, P. Nielsen, Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times, Appl. Math. Model. 65 (2019) 256–270.
[57] J. Dou, J. Li, C. Su, A novel feasible task sequence-oriented discrete particle swarm algorithm for simple assembly line balancing problem of type 1, Int. J. Adv. Manuf. Technol. 69 (2013) 2445–2457.
[58] L. Jian-sha, J. Ling-ling, L. Xiu-lin, Hybrid particle swarm optimization algorithm for assembly line balancing problem-2, in: 2009 16th International Conference on Industrial Engineering and Engineering Management, 2009.
[59] A. Al-Zuheri, L. Luong, K. Xing, Developing a multi-objective genetic optimisation approach for an operational design of a manual mixed-model assembly line with walking workers, J. Intell. Manuf. 27 (2016) 1049–1065.
[60] M. Alavidoost, M.F. Zarandi, M. Tarimoradi, Y. Nemati, Modified genetic algorithm for simple straight and U-shaped assembly line balancing with fuzzy processing times, J. Intell. Manuf. 28 (2017) 313–336.
[61] P. Fang, J. Yang, Q. Liao, R.Y. Zhong, Y. Jiang, Flexible worker allocation in aircraft final assembly line using multiobjective evolutionary algorithms, IEEE Trans. Ind. Inform. 17 (11) (2021) 7468–7478.
[62] R. Hwang, H. Katayama, A multi-decision genetic approach for workload balancing of mixed-model U-shaped assembly line systems, Int. J. Prod. Res. 47 (14) (2009) 3797–3822.
[63] R.K. Hwang, H. Katayama, M. Gen, U-shaped assembly line balancing problem with genetic algorithm, Int. J. Prod. Res. 46 (16) (2008) 4637–4649.
[64] S.M. Kazemi, R. Ghodsi, M. Rabbani, R. Tavakkoli-Moghaddam, A novel two-stage genetic algorithm for a mixed-model U-line balancing problem with duplicated tasks, Int. J. Adv. Manuf. Technol. 55 (2011) 1111–1122.
[65] Y.K. Kim, W.S. Song, J.H. Kim, A mathematical model and a genetic algorithm for two-sided assembly line balancing, Comput. Oper. Res. 36 (3) (2009) 853–865.
[66] A.C. Nearchou, Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization, Int. J. Prod. Econ. 129 (2) (2011) 242–250.
[67] J. Yu, Y. Yin, Assembly line balancing based on an adaptive genetic algorithm, Int. J. Adv. Manuf. Technol. 48 (2010) 347–354.
[68] Y. Zakharova, M. Sakhno, Complexity and heuristic algorithms for speed scaling scheduling of parallel jobs with energy constraint, J. Comput. Appl. Math. 457 (2025) 116254.
[69] W. Zhang, M. Gen, An efficient multiobjective genetic algorithm for mixed-model assembly line balancing problem considering demand ratio-based cycle time, J. Intell. Manuf. 22 (2011) 367–378.
[70] H. Babazadeh, A. Esfahanipour, A novel multi period mean-VaR portfolio optimization model considering practical constraints and transaction cost, J. Comput. Appl. Math. 361 (2019) 313–342.
[71] M. Rabbani, Z. Mousavi, H. Farrokhi-Asl, Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem, J. Ind. Prod. Eng. 33 (7) (2016) 472–484.

[72] Z. Li, M.N. Janardhanan, Q. Tang, Multi-objective migrating bird optimization algorithm for cost-oriented assembly line balancing problem with collaborative robots, Neural Comput. Appl. 33 (14) (2021) 8575–8596.
[73] J. Guo, Z. Pu, B. Du, Y. Li, Multi-objective optimisation of stochastic hybrid production line balancing including assembly and disassembly tasks, Int. J. Prod. Res. 60 (9) (2022) 2884–2900.
[74] Z. Xu, Y. Han, Z. Li, Y. Zou, Y. Chen, A New Pareto Discrete NSGAII Algorithm for Disassembly Line Balance Problem, Int. J. Intell. Syst. 2023 (1) (2023) 8847164.
[75] A. Farsi, M. Mokhtarzadeh, M. Rabbani, N. Manavizadeh, M. Ghasempour Anaraki, Using parallel metaheuristics to solve a parallel U-shaped robotic mixed-model assembly line balancing and sequencing problem, Soft Comput. 28 (21) (2024) 12603–12621.
[76] A. Scholl, N. Boysen, Designing parallel assembly lines with split workplaces: Model and optimization procedure, Int. J. Prod. Econ. 119 (1) (2009) 90–100.
[77] M. Rabbani, R. Siadatian, H. Farrokhi-Asl, N. Manavizadeh, Multi-objective optimization algorithms for mixed model assembly line balancing problem with parallel workstations, Cogent Eng. 3 (1) (2016) 1158903.
[78] Z.A. Cil, Z. Li, S. Mete, E. Özceylan, Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration, Appl. Soft Comput. 93 (2020) 106394.
[79] O. Yilmaz, N. Aydin, I. Kucukkoc, Integrated Model Assignment and Multi-Line Balancing in Human-Robot Collaborative Mixed-Model Assembly Lines (Working, Paper) (2025).
[80] N. Aydin, Z. Cetinkale, Analyses on ICU and non-ICU capacity of government hospitals during the COVID-19 outbreak via multi-objective linear programming: An evidence from Istanbul, Comput. Biol. Med. 146 (2022) 105562.
[81] N. Barathwaj, P. Raja, S. Gokulraj, Optimization of assembly line balancing using genetic algorithm, J. Cent. South Univ. 22 (2015) 3957–3969.
[82] I. Kucukkoc, S. Finco, M. Peron, G.A. Keskin, Including mechanical requirements in a bi-objective nesting and scheduling model for additive manufacturing, Eur. J. Oper. Res. (2025).
[83] Y. Zhou, Y. Peng, W. Li, D.T. Pham, Stackelberg model-based human-robot collaboration in removing screws for product remanufacturing, Robot. Comput. Integr. Manuf. 77 (2022) 102370.